

# The ifplatform package

Original code by Johannes Große

Package by Will Robertson

wspr81 at gmail dot com

vo.4\* 2010/10/22

## 1 Main features and usage

This package provides the three following conditionals to test which operating system is being used to run  $\TeX$ :

```
\ifwindows  
\iflinux  
\ifmacosx  
\ifcygwin
```

If you only wish to detect `\ifwindows`, then it does not matter how you load this package. Note then that use of (Linux *or* Mac OS X *or* Cygwin) can then be detected with `\ifwindows\else`.

If you also wish to determine the difference between which Unix-variant you are using (i.e., also detect `\iflinux`, `\ifmacosx`, and `\ifcygwin`) then shell escape must be enabled. This is achieved by using the `-shell-escape` command line option when executing  $\LaTeX$ .

If shell escape is not enabled, `\iflinux`, `\ifmacosx`, and `\ifcygwin` will all return *false*. A warning will be printed in the console output to remind you in this case.

---

\*Thanks to Ken Brown, Joseph Wright, Zebb Prime, and others for testing this version.

## 2 Auxiliary features

`\ifshellescape` is provided as a conditional to test whether shell escape is active or not. (Note: new versions of pdfTeX allow you to query shell escape with `\ifnum\pdfshellescape>0`, and the `pdftexcmds` provides the wrapper `\pdf@shellescape` which works with XeTeX, pdfTeX, and LuaTeX.)

Also, the `\platformname` command is defined to expand to a macro that represents the operating system. Default definitions are (respectively):

```
\windowsname      → 'Windows'
\notwindowsname   → '*NIX' (when shell escape is disabled)
\linuxname        → 'Linux'
\macosxname       → 'Mac OS X'
\cygwinname       → 'Cygwin'
\unknownplatform  → whatever is returned by uname
```

E.g., if `\ifwindows` is *true* then `\platformname` expands to `\windowsname`, which expands to 'Windows'. Redefine the macros above to customise the output of `\platformname`.

*This documentation was compiled on \*NIX.*

## 3 Other platforms

If greater granularity is required to differentiate between various UNIX-like operating systems, then `\unknownplatform` can be interrogated for the platform based on the output of `uname`. Table 1 lists possible outputs for a range of operating systems.

For example, to test whether the AIX operating system is being used, you could use the following code:

```
\def\myplatform{aix6}
\ifx\myplatform\unknownplatform
  ... AIX is being used ...
\else
  ... or not ...
\fi
```

The `ifthen` and `xifthen` packages might be of interest to those who prefer more L<sup>A</sup>T<sub>E</sub>X-like methods of conditional testing.

## 4 Limitations

Some technical information in case things go wrong.

- ifplatform checks for Windows by the presence or absence of the file 'nul:'. If you have a file in your search path in \*nix called 'nul:.tex' (or without the .tex) then things may become confused.
- ifplatform checks for \*nix by the presence or absence of the file '/dev/null'. If you have the file in Windows called /dev/null.tex (or without the extension) then things might similarly get mixed up.
- When both null files are detected (i.e., things aren't right with one of the two tests above), ifplatform uses another test to try and sort itself out. For interest, the test is: 'echo # > \jobname.w18'. Under Windows you should end up with a text file containing an octothorpe. On \*nix, the # will be seen as a comment char and the test will be ignored and the file will not be written.

This 'last resort' test will fail if shell escape is not enabled, or if the file \jobname.w18 somehow already exists, or if the behaviour of # isn't as reliable as I think.

- Note that if you're running T<sub>E</sub>X binaries from Cygwin on Windows, then your platform will *not* be Windows. It will appear to be a \*nix system, with platform name 'Cygwin'.
- If you ever see the error

```
I can't tell if this is Windows or *nix; you appear to  
be neither.
```

then I'd dearly like to know how it happened. It should never occur, as far as I know. (Update: in previous versions of this package, this message appeared when running under LuaT<sub>E</sub>X.)

Keep these points in mind and you'll never run into trouble. I hope you won't run into trouble in any case.

Platform	uname string
FreeBSD	FreeBSD
OpenBSD	OpenBSD
Solaris	SunOS
HPUX	HP-UX
IRIX	IRIX64
AIX	aix6
Cray UNICOS	sn5176

Table 1: List of operating systems and their uname strings. Adapted from <http://en.wikipedia.org/wiki/Uname>.

## 5 Implementation

```
1 \ProvidesPackage{ifplatform}
2 [2010/10/22 v0.4 Testing for the operating system]
```

Packages required: (thanks Heiko)

```
3 \RequirePackage{pdftexcmds,catchfile,ifluatex}
```

Conditionals we provide:

```
4 \newif\ifshellescape
5 \newif\ifwindows
6 \newif\ifmacosx
7 \newif\iflinux
8 \newif\ifcygwin
```

Names of operating systems:

```
\windowsname
notwindowsname
\linuxname
\macosxname
\cygwinname
unknownplatform
9 \newcommand\windowsname{Windows}
10 \newcommand\notwindowsname{*NIX}
11 \newcommand\linuxname{Linux}
12 \newcommand\macosxname{Mac\,OS\,X}
13 \newcommand\cygwinname{Cygwin}
14 \newcommand\unknownplatform{[Unknown]}
```

For internal stuff later:

```
15 \edef\ip@file{\jobname.w18}
16 \newif\if@ip@nix@
```

Determine if shell escape is enabled:

```
\ifshellescape
17 \ifnum\pdf@shellescape=1\relax
18 \shellescapetrue
19 \else
20 \ifluatex\else
21 \PackageWarningNoLine{ifplatform}{^^J \space\space\space
22 shell escape is disabled,
23 so I can only detect \@backslashchar ifwindows%
24 }
25 \fi
26 \fi
```

An error message for when things go wrong:

```
27 \def\ip@cantdecide{%
28 \PackageWarningNoLine{ifplatform}{^^J \space\space\space
29 I can't tell if this is Windows or *nix;
```

File	Exists	Windows?	*nix?
nul:	true	Probably	Maybe
	false	Definitely not	Definitely
/dev/null	true	Maybe	Probably
	false	Definitely	Definitely not

Table 2: Possibilities for testing null files and their prospects for determining the platform.

```

30     you appear to be both%
31   }%
32 }

```

Now the platform test. In LuaTeX this is straightforward:

```

33 \ifluatex
34   \csname\directlua{
35     if os.type == "unix" then
36       tex.sprint("@ip@nix@true")
37     elseif os.type == "windows" then
38       tex.sprint("windowstrue")
39     end
40   }\endcsname
41 \else

```

Otherwise we need to test for the null files of Windows and \*nix. (This doesn't work at all in LuaTeX. Not sure why; haven't looked.) In a normal situation, this is all we need to do:

```

42 \IfFileExists{nul:}{\@ip@nix@false}{\@ip@nix@true}
43 \IfFileExists{/dev/null}{\windowfalse}{\windowstrue}

```

However, sometimes that's not good enough. If things go wrong above, we still don't know which platform. Can only proceed if shell escape is on; fallback heuristic:

- If the tmp file exists
  - Tell them to delete it and abort.
  - Otherwise:
- Write to it with echo that only works on Windows
- Then see again if it exists
  - If the tmp file exists: Windows (and delete the file)
  - Otherwise: \*nix

Here's the code for the above 'last resort' test:

```
44 \edef\ip@windows@echo@test{echo \string# > "\ip@file"}
45 \def\ip@backupplan{%
46   \IfFileExists{\ip@file}{%
47     \PackageWarningNoLine{ifplatform}{^^J \space\space\space
48       Please delete the file "\ip@file" and try again%
49     }%
50     \ip@cantdecide
51   }{%
52     \immediate\write18{\ip@windows@echo@test}%
53     \IfFileExists{\ip@file}{%
54       \immediate\write18{del "\ip@file"}%
55       \windowstrue
56     }{%
57       \ip@nix@true
58     }%
59   }%
60 }
```

Now we use some odd logic to deduce what's happening in the edge cases when things go wrong: (see table 2)

```
61 \ifwindows
62   \if@ip@nix@
63     \PackageWarningNoLine{ifplatform}{^^J \space\space\space
64       I can't tell if this is Windows or *nix;
65       you appear to be neither%
66     }
67   \fi
68 \else
69   \if@ip@nix@\else
70     \ifshellescape
71       \ip@backupplan
72     \else
73       \ip@cantdecide
74     \fi
75   \fi
76 \fi
77 \fi
```

Needed below:

```
78 \def\ip@only@six#1#2#3#4#5#6#7\@nil{#1#2#3#4#5#6}
```

`\iflinux` Now test for the others; directly test for Linux and MacOSX; but what about  
`\ifmacosx` Solaris or FreeBSD or ... ? Define `\unknownplatform` as the output of `uname`  
`\ifcygwin` rather than enumerate the possibilities.

```

79 \if@ip@nix@
80 \ifshellescape
81 \ifwindows\else
82 \immediate\write18{uname -s > "\ip@file"}
83 \CatchFileDef\@tempa{\ip@file}{}
84 \immediate\write18{rm -- "\ip@file"}

```

Kill a trailing space:

```

85 \edef\@tempa{\expandafter\zap@space\@tempa\@empty}
86 \def\@tempb{Linux}
87 \ifx\@tempa\@tempb
88 \linuxtrue
89 \else
90 \def\@tempb{Darwin}
91 \ifx\@tempa\@tempb
92 \macosxtrue
93 \else
94 \def\@tempb{CYGWIN}
95 \edef\@tempc{\expandafter\ip@only@six\@tempa-----\@nil}
96 \ifx\@tempb\@tempc
97 \cygwintrue
98 \else
99 \edef\unknownplatform{\@tempa}
100 \fi
101 \fi
102 \fi
103 \fi
104 \fi\fi

```

`\platformname` Defined in terms of macros so the output is user-customisable.

```

105 \edef\platformname{%
106 \ifwindows
107 \noexpand\windowsname
108 \else
109 \ifshellescape
110 \iflinux
111 \noexpand\linuxname
112 \else

```

```

113     \ifmacosx
114         \noexpand\macosxname
115     \else
116         \ifcygwin
117             \noexpand\cygwinname
118         \else
119             \noexpand\unknownplatform
120         \fi
121     \fi
122 \fi
123 \else
124     \noexpand\notwindowsname
125 \fi
126 \fi
127 }

```

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols	D	I
\, . . . . . 12	\def . 27, 45, 78, 86, 90, 94	\if@ip@nix@ 16, 62, 69, 79
\@backslashchar . . . . 23	\directlua . . . . . 34	\ifcygwin . . . . 8, <u>79</u> , 116
\@empty . . . . . 85		\IfFileExists . . . .
\@ip@nix@false . . . . 42		. . . . 42, 43, <u>46</u> , 53
\@ip@nix@true . . . 42, 57	<b>E</b>	\iflinux . . . . 7, <u>79</u> , 110
\@nil . . . . . 78, 95	\edef 15, 44, 85, 95, 99, 105	\ifluatex . . . . . 20, 33
\@tempa . . . . .	\else . . . . . 19, 20,	\ifmacosx . . . . 6, <u>79</u> , 113
83, 85, 87, 91, 95, 99	41, 68, 69, 72, 81,	\ifnum . . . . . 17
\@tempb . . . . .	89, 93, 98, 108,	\ifshellescape . . . .
86, 87, 90, 91, 94, 96	112, 115, 118, 123	. . 4, <u>17</u> , 70, 80, 109
\@tempc . . . . . 95, 96	\endcsname . . . . . 40	\ifwindows . 5, 61, 81, 106
	\expandafter . . . . 85, 95	\ifx . . . . . 87, 91, 96
<b>C</b>		\immediate . 52, 54, 82, 84
\CatchFileDef . . . . . 83	<b>F</b>	\ip@backupplan . . 45, 71
\csname . . . . . 34	\fi . . . . . 25, 26, 67,	\ip@cantdecide 27, 50, 73
\cygwinname . . . . . <u>9</u> , 117	74–77, 100–104,	\ip@file . . . . 15, 44,
\cygwintrue . . . . . 97	120–122, 125, 126	

46, 48, 53, 54, 82–84	\newif . . . . . 4–8, 16	<b>S</b>
\ip@only@six . . . . 78, 95	\noexpand . . 107, 111,	\shellescapetrue . . . 18
\ip@windows@echo@test	114, 117, 119, 124	\space . . . . 21, 28, 47, 63
. . . . . 44, 52	\notwindowsname . <u>9</u> , 124	\string . . . . . 44
<b>J</b>		<b>U</b>
\jobname . . . . . 15	<b>P</b>	\unknownplatform . .
<b>L</b>	\PackageWarningNoLine	. . . . . <u>9</u> , 99, 119
\linuxname . . . . . <u>9</u> , 111	. . . . 21, 28, 47, 63	<b>W</b>
\linuxtrue . . . . . 88	\pdf@shellescape . . . 17	\windowsfalse . . . . . 43
<b>M</b>	\platformname . . . . <u>105</u>	\windowsname . . . . <u>9</u> , 107
\macosxname . . . . . <u>9</u> , 114	\ProvidesPackage . . . . 1	\windowstrue . . . . 43, 55
\macosxtrue . . . . . 92	<b>R</b>	\write . . . . 52, 54, 82, 84
<b>N</b>	\relax . . . . . 17	<b>Z</b>
\newcommand . . . . . 9–14	\RequirePackage . . . . 3	\zap@space . . . . . 85