

# texosquery: query OS information from T<sub>E</sub>X

Nicola L. C. Talbot

<http://www.dickimaw-books.com/>

2016-07-08 (v1.0)

## Abstract

`texosquery.jar` is a cross-platform Java application to query certain OS information designed for use in T<sub>E</sub>X's shell escape mechanism. The accompanying T<sub>E</sub>X code provides a simple interface to the `texosquery` Java application, which may be skipped if you want to explicitly use `\input` (but take care of any special characters appearing in the result).

**Important Note:** You will need T<sub>E</sub>X's shell escape enabled, and you will also need the Java Runtime Environment (JRE) installed to use `texosquery`.

## Contents

<b>1</b>	<b><code>texosquery.jar</code>: the Java application</b>	<b>1</b>
<b>2</b>	<b><code>texosquery.tex</code>: generic T<sub>E</sub>X code</b>	<b>3</b>
<b>3</b>	<b>The Code</b>	<b>6</b>
3.1	Generic T <sub>E</sub> X Code . . . . .	6
3.2	L <sup>A</sup> T <sub>E</sub> X Code . . . . .	10
	<b>Change History</b>	<b>10</b>
	<b>Index</b>	<b>10</b>

## 1 `texosquery.jar`: the Java application

The `texosquery` Java command line application looks up certain system information that may be of use in T<sub>E</sub>X documents. All this information can easily be obtained using native commands, but the Java application allows an OS-independent approach with results that can easily be captured by T<sub>E</sub>X's shell-escape without having to strip formatting information.

**Important Note:** `texosquery` provides read-only actions, and I don't intend adding any actions that modify system settings or files.

Since the application is designed to work with T<sub>E</sub>X, each function will display the result on a single line without formatting. A blank line will be displayed if the information isn't available. A forward slash is always used as a directory divider, regardless of the operating system, so the result can be used, for example, in `\input` or `\includegraphics`.

If an input file name is required (for example, with the `--pdfdate` argument described below) then the file may be in the current working directory, relative to the current directory (with forward slash `/` as the directory divider), an absolute path (again with forward slash) or on T<sub>E</sub>X's path (in which case, `kpsewhich` is used to locate it).

Command line invocation:

```
texosquery <action> ...
```

Available actions (at least one required):

**-L or --locale** Display the locale information in the form

`<lang>-<region>.<codeset>@<modifier>`

where `<lang>` is the ISO code for the language (e.g. `en`), `<region>` is the ISO code for the region (e.g. `GB`), `<codeset>` is the default code set (e.g. `UTF-8`) and `<modifier>` is the modifier. Elements may be omitted if unavailable. For example, `en-GB.UTF-8` has the `<modifier>` omitted, and `en` has all but the language omitted.

**-l or --locale-lcs** As the above but the codeset (if present) is converted to lower case and any hyphens are stripped. For example, if `--locale` returns `en-GB.UTF-8`, then `--locale-lcs` would return `en-GB.utf8`.

**-c or --cwd** Displays the current working directory.

**-m or --userhome** Displays the user's home directory.

**-t or --tmpdir** Displays the temporary directory.

**-o or --osname** Displays the operating system name.

**-r or --osversion** Displays the operating system version.

**-a or --osarch** Displays the operating system architecture.

**-n or --pdfnow** Displays the current date and time in PDF format. For example

```
D:20160704131006+01'00'
```

Note that some, but not all, T<sub>E</sub>X formats provide `\pdfcreationdate`, which is more efficient than using the shell escape, but this can be used as a fallback method for those that don't (for example, X<sub>Y</sub>T<sub>E</sub>X).

**-d <file> or --pdfdate <file>** Displays the last modified time stamp of the given file in PDF format or a blank line if the file doesn't exist or the file permissions prohibit this action. Again some, but not all, T<sub>E</sub>X formats provide `\pdffilemoddate{<file>}`, which is more efficient than using the shell escape.

- s *<file>* or --filesize *<file>* Displays the size in bytes of the given file or a blank line if the file doesn't exist or the file permissions prohibit this action. Some, but not all, T<sub>E</sub>X formats provide `\pdffilesize{<file>}`, which is more efficient than using the shell escape.
- i *<sep>* *<dir>* or --list *<sep>* *<dir>* List all files in the given directory with the output on a single line using *<sep>* as the separator between entries. Note that the list doesn't include the full path, just the file names.
- f *<sep>* *<regex>* *<dir>* or --filterlist *<sep>* *<regex>* *<dir>* Like the above but only lists those files whose name matches the regular expression given in *<regex>*. Note that this tests for a complete match on the file name (not including path). For example, if *<regex>* is `foo.*`, it will only match files whose name starts with `foo` (for example, `foobar` will match but `barfoo` won't).
- u *<file>* or --uri *<file>* Displays the URI of the given file or a blank line if the file doesn't exist or the file permissions prohibit this action. Note that this may include in a percent character in the result. The T<sub>E</sub>X command `\TeXOSQuery` protects against this by changing the category code, but if you explicitly call `texosquery` using the shell escape, you'll need to be careful of this.
- p *<file>* or --path *<file>* Displays the canonical path of the given file or a blank line if the file doesn't exist or the file permissions prohibit this action.
- h or --help Displays help message and exits.
- v or --version Displays version information and exits.

If multiple options are given, they will be processed in the order specified in the command line invocation. Each result will be displayed on a separate line.

## 2 texosquery.tex: generic T<sub>E</sub>X code

You can run `texosquery` directly from T<sub>E</sub>X's shell escape. For example:

```
\input|"texosquery --locale"
```

However, `texosquery.tex` provides generic T<sub>E</sub>X code to do this for you and store the result in a control sequence.

Plain T<sub>E</sub>X users can input this file through the usual `\input` method:

```
\input texosquery
```

L<sup>A</sup>T<sub>E</sub>X users may also simply input this file:

```
\input{texosquery}
```

but may prefer the standard package approach:

```
\usepackage{texosquery}
```

**Important Note:** The commands described below are all fragile.

The basic command to run `texosquery` and capture its output in a control sequence is:

`\TeXOSQuery{<cs>}{<args>}`

where `<cs>` is the control sequence in which to store the result and `<args>` are the command line arguments to pass to `texosquery`. This first locally changes the category code of some problematic characters that may appear in the result. The only special characters that aren't changed are the backslash `\`, curly braces `{` and `}`, and hash `#`. (`texosquery.jar` will replace `#` with `\#` in places where it might possibly occur in the result, but in general it's best to avoid these characters in file names.) There are some short cut commands for convenience, described below.

If any of these commands cause an error message in the form:

I can't find file '`|texosquery`'.

then check that you have the shell escape on. If the error persists with the shell escape enabled (and not restricted), check that `texosquery` is on your system's path. To do this, open a command prompt or terminal and type `texosquery`. If it isn't installed correctly, there will be a message like:

`'texosquery' is not recognised`

or

`texosquery: command not found`

If this happens, check the installation. (Instructions are in the accompanying `README.md` file.)

`\ifTeXOSQueryDryRun` Dry run mode is determined by the conditional

`\ifTeXOSQueryDryRun`

If true, the shell escape won't be used and the requested command invocation will be printed in the transcript file prefixed with

`TeXOSQuery:`

(the control sequence `<cs>` will be set to empty). This conditional will automatically be switched on if `\shellescape` or `\pdfshellescape` is 0.

`\TeXOSQueryLocale` The locale (`-l` or `--locale-lcs`) information can be obtained using:

`\TeXOSQueryLocale{<cs>}`

Note that this uses the lower case codeset form, which has a better chance of matching the encoding names used by the `inputenc` package. If you want the unprocessed codeset name, you can do:

`\TeXOSQuery{<cs>}{-L}`

`\TeXOSQueryCwd`      The current working directory (`-c` or `--cwd`) can be obtained using:

`\TeXOSQueryCwd{<cs>}`

`\TeXOSQueryHome`      The home directory (`-m` or `--userhome`) can be obtained using:

`\TeXOSQueryHome{<cs>}`

`\TeXOSQueryTmpDir`      The temporary directory (`-t` or `--tmpdir`) can be obtained using:

`\TeXOSQueryTmpDir{<cs>}`

`\TeXOSQueryName`      The OS name (`-o` or `--osname`) can be obtained using:

`\TeXOSQueryName{<cs>}`

`\TeXOSQueryVersion`      The OS version (`-r` or `--osversion`) can be obtained using:

`\TeXOSQueryVersion{<cs>}`

`\TeXOSQueryArch`      The OS architecture (`-a` or `--osarch`) can be obtained using:

`\TeXOSQueryArch{<cs>}`

The current date-time stamp in PDF format (`-n` or `--pdfnow`) can be obtained using:

`\TeXOSQueryNow`

`\TeXOSQueryNow{<cs>}`

This is provided for the benefit of users who don't have `\pdfcreationdate` defined by their  $\TeX$  format (for example,  $\text{\LaTeX}$ ).

**Important Note:** The remaining commands all require extra arguments after the relevant switch. These are automatically enclosed in single-quotes to protect any spaces. If the argument actually contains any single-quote characters, make sure you use `\string\'` to prevent interference. *However, in general it's a very bad idea to use quotes as part of a file name (rather than using them as delimiters).* Since the file name may need to be obtained from `\jobname`, which sometimes includes double-quotes, the first double-quote pair found is stripped in file name arguments. Any other double-quotes will need to be protected in the same manner as single-quotes (but, again, this shouldn't be an issue if you use a safe file naming scheme). Any paths should use a forward slash for the directory divider.

`\TeXOSQueryFileDate`      The modification date-time stamp in PDF format for a file (`-d` or `--pdfdate`) can be obtained using:

`\TeXOSQueryFileDate{<cs>}{<filename>}`

where `<filename>` is the name of the file. This is provided for the benefit of users who don't have `\pdffilemoddate` defined by their  $\TeX$  format.

`\TeXOSQueryFileSize`      The size in bytes of a file (`-s` or `--filesize`) can be obtained using:

`\TeXOSQueryFileSize{<cs>}{<filename>}`

where *<filename>* is the name of the file. This is provided for the benefit of users who don't have `\pdffilesize` defined by their  $\TeX$  format.

`\TeXOSQueryFileList`      The list of files in a given directory (`-i` or `--list`) can be obtained using:

`\TeXOSQueryFileList{<cs>}{<sep>}{<dir>}`

where *<sep>* is the separator and *<dir>* is the directory name. For example:

`\TeXOSQueryFileList{\result}{,}{.}`

will store a comma-separated list of all the files contained in the current directory in the control sequence `\result`.

`\TeXOSQueryFilterFileList`      A filtered list of files in a given directory (`-f` or `--filterlist`) can be obtained using:

`\TeXOSQueryFilterFileList{<cs>}{<sep>}{<regex>}{<dir>}`

where *<regex>* is a regular expression. *Take care of any backslashes in the regular expression!* For example, to list only those files that have an extension:

`\TeXOSQueryFilterFileList{\result}{,}{.+\\string\\. *}{.}`

Note the use of `\\string\.` to ensure that `\.` isn't interpreted as a command. Another example, list only `.png` and `.jpg` files in the directory called `images`:

`\TeXOSQueryFilterFileList{\result}{,}{.+\\string\\. (jpg|png)}{images}`

`\TeXOSQueryFileURI`      The URI of a file (`-u` or `--uri`) can be obtained using:

`\TeXOSQueryFileURI{<cs>}{<filename>}`

where *<filename>* is the name of the file. (Any percent symbols `%` contained in the URI will have their category code set to 12.)

`\TeXOSQueryFilePath`      The canonical path of a file (`-p` or `--path`) can be obtained using:

`\TeXOSQueryFilePath{<cs>}{<filename>}`

where *<filename>* is the name of the file.

## 3 The Code

### 3.1 Generic $\TeX$ Code

Change category code of `@` if necessary.

```
1 \ifnum\catcode'\@=11\relax
2   \def\@texosquery@restore@at{}%
3 \else
4   \expandafter\edef\csname @texosquery@restore@at\endcsname{%
5     \noexpand\catcode'\noexpand\@=\number\catcode'\@ \relax
6   }%
7   \catcode'\@=11\relax
8 \fi
```

Check if already loaded.

```
9 \ifx\TeXOSQuery\undefined
10 \else
11 \@texosquery@restore@at
12 \expandafter\endinput
13 \fi
```

\TeXOSInvokerName The name of the texosquery application.

```
14 \def\TeXOSInvokerName{texosquery}
```

\TeXOSQueryInvoker If we're using L<sup>A</sup>T<sub>E</sub>X, we'll need to use \@@input rather than \input.

```
15 \ifx\@@input\undefined
16 \def\TeXOSQueryInvoker#1{\input|" \TeXOSInvokerName\space#1" }
17 \else
18 \def\TeXOSQueryInvoker#1{\@@input|" \TeXOSInvokerName\space#1" }
19 \fi
```

\ifTeXOSQueryDryRun Provide a dry-run mode.

```
20 \newif\ifTeXOSQueryDryRun
21 \TeXOSQueryDryRunfalse
```

If shell escape is disabled, automatically switch on dry-run mode.

```
22 \ifx\shellescape\undefined
23 \ifx\pdfshellescape\undefined
24 \else
25 \ifnum\pdfshellescape=0\relax
26 \TeXOSQueryDryRuntrue
27 \fi
28 \fi
29 \else
30 \ifnum\shellescape=0\relax
31 \TeXOSQueryDryRuntrue
32 \fi
33 \fi
```

\TeXOSQuery Use texosquery with the option given in the second argument and store the result in control sequence given in the first argument.

```
34 \def\TeXOSQuery#1#2{%
35 \ifTeXOSQueryDryRun
36 \begingroup
37 \newlinechar='^^J
38 \message{^^JTeXOSQuery: \TeXOSInvokerName\space#2^^J}%
39 \endgroup
40 \def#1{}%
41 \else
42 \begingroup
43 \endlinechar=-1\relax
```

Just in case the result contains any awkward characters that have some special meaning to  $\TeX$ . (Can't really protect against hash, backslash or curly braces, but they're unlikely to occur unless the user has a very wacky and inappropriate file naming scheme.)

```

44 \catcode'\-=12\relax
45 \catcode'\_ =12\relax
46 \catcode'\^ =12\relax
47 \catcode'\~ =12\relax
48 \catcode'\$=12\relax
49 \catcode'\&=12\relax
50 \catcode'\ "=12\relax
51 \catcode'\ ' =12\relax
52 \catcode'\.=12\relax
53 \catcode'\/=12\relax
54 \catcode'\:=12\relax
55 \catcode'\;=12\relax
56 \catcode'\%=12\relax
57 \everyeof{\noexpand}\relax
58 \edef\x{\endgroup\def\noexpand#1{\TeXOSQueryInvoker{#2}}}\x
59 \fi
60 }

```

Now for some convenient shortcuts so the user doesn't have to remember the command line options. `\string` is used in case the hyphen character has been made active.

<code>\TeXOSQueryLocale</code>	Query the locale and store the result in the control sequence provided in the argument. 61 \def \TeXOSQueryLocale#1{\TeXOSQuery{#1}{\string-l}}
<code>\TeXOSQueryCwd</code>	Query the current working directory. 62 \def \TeXOSQueryCwd#1{\TeXOSQuery{#1}{\string-c}}
<code>\TeXOSQueryHome</code>	Query the user's home directory. 63 \def \TeXOSQueryHome#1{\TeXOSQuery{#1}{\string-m}}
<code>\TeXOSQueryTmpDir</code>	Query the temporary directory. 64 \def \TeXOSQueryTmpDir#1{\TeXOSQuery{#1}{\string-t}}
<code>\TeXOSQueryVersion</code>	Query the operating system version. 65 \def \TeXOSQueryVersion#1{\TeXOSQuery{#1}{\string-r}}
<code>\TeXOSQueryArch</code>	Query the operating system architecture. 66 \def \TeXOSQueryArch#1{\TeXOSQuery{#1}{\string-a}}
<code>\TeXOSQueryName</code>	Query the operating system name. 67 \def \TeXOSQueryName#1{\TeXOSQuery{#1}{\string-o}}
<code>\TeXOSQueryNow</code>	Query the current time stamp. 68 \def \TeXOSQueryNow#1{\TeXOSQuery{#1}{\string-n}}



If the file name is supplied using `\jobname` it may have double-quotes which will interfere with things.

`\texosquerystripquotes`

```
69 \def\texosquerystripquotes#1{%
70   \@texosquery@stripquotes#1\@mid\texosquery@stripquotes
71   "\relax"\relax\@end\texosquery@stripquotes
72 }
73 \def\@texosquery@stripquotes#1"#2"{%
74   \@@texosquery@stripquotes#1#2%
75 }
76 \def\@@texosquery@stripquotes#1\@mid\texosquery@stripquotes#2\@end\texosquery@stripquotes{%
77   #1%
78 }
```

`\TeXOSQueryFileDate` Query the time stamp of the file given in the second argument.

```
79 \def\TeXOSQueryFileDate#1#2{\TeXOSQuery{#1}{\string-d
80   \string'\texosquerystripquotes{#2}\string'}}
```

`\TeXOSQueryFileSize` Query the size of the file given in the second argument.

```
81 \def\TeXOSQueryFileSize#1#2{\TeXOSQuery{#1}{\string-s
82   \string'\texosquerystripquotes{#2}\string'}}
```

`\TeXOSQueryFileList` List all files in the directory given in the third argument, separated by the second argument.

```
83 \def\TeXOSQueryFileList#1#2#3{\TeXOSQuery{#1}{%
84   \string-i \string'#2\string'
85   \string'\texosquerystripquotes{#3}\string'}}
```

`\TeXOSQueryFilterFileList` Filtered list files in the directory given in the fourth argument, separated by the second argument. The third argument is the regular expression used to filter the list. *Take care of backslashes in the regular expression!*

```
86 \def\TeXOSQueryFilterFileList#1#2#3#4{\TeXOSQuery{#1}{%
87   \string-f \string'#2\string' \string'#3\string'
88   \string'\texosquerystripquotes{#4}\string'}}
```

`\TeXOSQueryFileURI` Get the URI of the file given in the second argument.

```
89 \def\TeXOSQueryFileURI#1#2{\TeXOSQuery{#1}{\string-u
90   \string'\texosquerystripquotes{#2}\string'}}
```

`\TeXOSQueryFilePath` Get the canonical path of the file given in the second argument.

```
91 \def\TeXOSQueryFilePath#1#2{\TeXOSQuery{#1}{\string-p
92   \string'\texosquerystripquotes{#2}\string'}}
```

All done. Restore the category code of `@`:

```
93 \@texosquery@restore@at
```

## 3.2 L<sup>A</sup>T<sub>E</sub>X Code

This is just a simple wrapper for `texosquery.tex` so that it can be loaded using L<sup>A</sup>T<sub>E</sub>X's standard `\usepackage` method. Identify package:

```
94 \NeedsTeXFormat{LaTeX2e}
95 \ProvidesPackage{texosquery}[2016/07/08 v1.0 (NLCT)]
```

Load `texosquery.tex`:

```
96 \input{texosquery}
```

That's it!

## Change History

1.0

General: Initial release . . . . . 6

<b>I</b>		
<code>\ifTeXOSQueryDryRun</code>	4, 7	
<b>T</b>		
<code>\TeXOSInvokerName</code>	7	
<code>\TeXOSQuery</code>	4, 7	
texosquery options		
<code>-a</code>	2, 5	
<code>-c</code>	2, 5	
<code>--cwd</code>	2, 5	
<code>-d</code>	2, 5	
<code>-f</code>	3, 6	
<code>--filesize</code>	3, 5	
<code>--filterlist</code>	3, 6	
<code>-h</code>	3	
<code>--help</code>	3	
<code>-i</code>	3, 6	
<code>-L</code>	2, 4	
<code>-l</code>	2, 4	
<code>--list</code>	3, 6	
<code>--locale</code>	2	
<code>--locale-lcs</code>	2, 4	<code>\TeXOSQueryCwd</code> . . . . 5, 8
<code>-m</code>	2, 5	<code>\TeXOSQueryFileDate</code>
<code>-n</code>	2, 5	. . . . . 5, 9
<code>-o</code>	2, 5	<code>\TeXOSQueryFileList</code>
<code>--osarch</code>	2, 5	. . . . . 6, 9
<code>--osname</code>	2, 5	<code>\TeXOSQueryFilePath</code>
<code>--osversion</code>	2, 5	. . . . . 6, 9
<code>-p</code>	3, 6	<code>\TeXOSQueryFileSize</code>
<code>--path</code>	3, 6	. . . . . 5, 9
<code>--pdfdate</code>	2, 5	<code>\TeXOSQueryFileURI</code> 6, 9
<code>--pdfnow</code>	2, 5	<code>\TeXOSQueryFilterFileList</code>
<code>-r</code>	2, 5	. . . . . 6, 9
<code>-s</code>	3, 5	<code>\TeXOSQueryHome</code> . . . 5, 8
<code>-t</code>	2, 5	<code>\TeXOSQueryInvoker</code> . . 7
<code>--tmpdir</code>	2, 5	<code>\TeXOSQueryLocale</code> . 4, 8
<code>-u</code>	3, 6	<code>\TeXOSQueryName</code> . . . 5, 8
<code>--uri</code>	3, 6	<code>\TeXOSQueryNow</code> . . . . 5, 8
<code>--userhome</code>	2, 5	<code>\texosquerystripquotes</code>
<code>-v</code>	3	. . . . . 9
<code>--version</code>	3	<code>\TeXOSQueryTmpDir</code> . 5, 8
<code>\TeXOSQueryArch</code>	5, 8	<code>\TeXOSQueryVersion</code> 5, 8