texosquery: query OS information from TEX

Nicola L. C. Talbot
dickimaw-books.com

Paulo Cereda

2016-07-14 (v1.1)

Abstract

`texosquery.jar` is a cross-platform Java application to query certain OS information designed for use in TEX's shell escape mechanism. The accompanying TEX 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 TEX's shell escape enabled, and you will also need the Java Runtime Environment (JRE) installed to use `texosquery`.

If you want to rebuild the application, instructions for compiling the source code (including the code for this document) are in the accompanying `README.md` file.

Contents

1 texosquery.jar: the Java application

The `texosquery` Java command line application looks up certain system information that may be of use in TEX 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 TEX'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 TEX, each function will display the result on a single line without formatting. (For multiple results, each line is grouped from v1.1.) A blank line (or empty group) 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 TEX'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, TEX 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, XƎTEX).

`-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, TEX 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, TEX 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 ⟨*regexp*⟩ 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 TEX 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.

`-e` ⟨***file***⟩ **or** `--dirname` ⟨***file***⟩ (New to v1.1.) Displays the canonical path of the given file's parent (that is, the directory containing ⟨*file*⟩) or a blank line if the file doesn't exist or the file permissions prohibit this action. Note that this is different to the Unix-like `dirname` command, which will return a relative path if ⟨*file*⟩ isn't an absolute path.

`-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. As from v1.1, if there are multiple actions, each result will be grouped. This makes it easier to process the results in TEX. For example:

```
texosquery -l
```

This just produces (for me):

```
en-GB.utf8
```

whereas

```
texosquery -l -n
```

produces:

```
{en-GB.utf8}
{D:20160714112732+01'00'}
```

Note that unavailable information will produce an empty group. For example (assuming `nofile` doesn't exist):

```
texosquery -l -d nofile -n
```

produces:

```
{en-GB.utf8}
{}
{D:20160714112732+01'00'}
```

whereas

```
texosquery -d nofile
```

just displays an empty line.

2 texosquery.tex: generic TEX code

You can run `texosquery` directly from TEX's shell escape. For example:

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

However, `texosquery.tex` provides generic TEX code to do this for you and store the result in a control sequence.

Plain TEX users can input this file through the usual `\input` method:

```
\input texosquery
```

LATEX 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.

`\TeXOSQuery` 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 unless `\shellescape` or `\pdfshellescape` is 1. (If `texosquery.jar` is later allowed on the restricted list, newer versions may change this default.)

If multiple queries are required, it's more efficient to perform them all in one go. For example:

```
\TeXOSQuery{\result}{-l -n -o}

\def\parseresult#1#2#3{%
  Locale: #1. Now: #2. OS: #3.%
}

\ifx\result\empty
  Query failed.
\else
  \expandafter\parseresult\result
\fi
```

(Make sure you have at least v1.1 for this to work correctly.)

\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*⟩`}`

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

`\TeXOSQueryNow{`⟨*cs*⟩`}`

This is provided for the benefit of users who don't have `\pdfcreationdate` defined by their TEX format (for example, XETEX). As from v1.1, this ensures that the initial `D` has category code 12 (which won't happen if `\TeXOSQuery` is used explicitly).

> **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.

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

`\TeXOSQueryFileDate`

`\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. As from v1.1, this ensures that the initial `D` has category code 12 (which won't happen if `\TeXOSQuery` is used explicitly).

`\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.

`\TeXOSQueryDirName`

The canonical path of a file's parent (`-e` or `--dirname`) can be obtained using:

`\TeXOSQueryDirName{`⟨*cs*⟩`}{`⟨*filename*⟩`}`

where ⟨*filename*⟩ is the name of the file.

3 The Code

3.1 Generic TEX Code

Change category code of `@` if necessary.

```
\ifnum\catcode`\@=11\relax
  \def\@texosquery@restore@at{}%
\else
  \expandafter\edef\csname @texosquery@restore@at\endcsname{%
    \noexpand\catcode`\noexpand\@=\number\catcode`\@\relax
  }%
 \catcode`\@=11\relax
\fi
```

Check if already loaded.

```
\ifx\TeXOSQuery\undefined
\else
  \@texosquery@restore@at
  \expandafter\endinput
\fi
```

`\TeXOSInvokerName` The name of the `texosquery` application.

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

`\TeXOSQueryInvoker` If we're using LATEX, we'll need to use `\@@input` rather than `\input`.

```
\ifx\@@input\undefined
  \def\TeXOSQueryInvoker#1{\input|"\TeXOSInvokerName\space#1" }
\else
  \def\TeXOSQueryInvoker#1{\@@input|"\TeXOSInvokerName\space#1" }
\fi
```

`\ifTeXOSQueryDryRun` Provide a dry-run mode.

```
\newif\ifTeXOSQueryDryRun
\TeXOSQueryDryRuntrue
```

If shell escape is unrestricted, automatically switch off dry-run mode.

```
\ifx\shellescape\undefined
  \ifx\pdfshellescape\undefined
  \else
    \ifnum\pdfshellescape=1\relax
      \TeXOSQueryDryRunfalse
    \fi
  \fi
\else
  \ifnum\shellescape=1\relax
    \TeXOSQueryDryRunfalse
  \fi
\fi
```

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

```
\def\TeXOSQuery#1#2{%
  \ifTeXOSQueryDryRun
    \begingroup
      \newlinechar=`\^^J
      \message{^^JTeXOSQuery: \TeXOSInvokerName\space#2^^J}%
    \endgroup
    \def#1{}%
  \else
    \begingroup
    \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.)

```
    \catcode`\-=12\relax
    \catcode`\_=12\relax
    \catcode`\^=12\relax
    \catcode`\~=12\relax
    \catcode`\$=12\relax
    \catcode`\&=12\relax
    \catcode`\"=12\relax
    \catcode`\'=12\relax
    \catcode`\.=12\relax
    \catcode`\/=12\relax
    \catcode`\:=12\relax
    \catcode`\;=12\relax
    \catcode`\%=12\relax
    \everyeof{\noexpand}\relax
    \edef\x{\endgroup\def\noexpand#1{\TeXOSQueryInvoker{#2}}}\x
  \fi
}
```

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.

`\TeXOSQueryLocale` Query the locale and store the result in the control sequence provided in the argument.

```
\def\TeXOSQueryLocale#1{\TeXOSQuery{#1}{\string-l}}
```

`\TeXOSQueryCwd` Query the current working directory.

```
\def\TeXOSQueryCwd#1{\TeXOSQuery{#1}{\string-c}}
```

`\TeXOSQueryHome` Query the user's home directory.

```
\def\TeXOSQueryHome#1{\TeXOSQuery{#1}{\string-m}}
```

`\TeXOSQueryTmpDir` Query the temporary directory.

```
\def\TeXOSQueryTmpDir#1{\TeXOSQuery{#1}{\string-t}}
```

\TeXOSQueryVersion Query the operating system version.

```
\def\TeXOSQueryVersion#1{\TeXOSQuery{#1}{\string-r}}
```

\TeXOSQueryArch Query the operating system architecture.

```
\def\TeXOSQueryArch#1{\TeXOSQuery{#1}{\string-a}}
```

\TeXOSQueryName Query the operating system name.

```
\def\TeXOSQueryName#1{\TeXOSQuery{#1}{\string-o}}
```

\TeXOSQueryNow Query the current time stamp.

```
\def\TeXOSQueryNow#1{%
```

The `D` needs category code 12. (Don't need to worry about `Z` as `texosquery.jar` uses `+00'00'` for UTC+0.) This change can't be done with the other catcode changes in `\TeXOSQuery`, as this is only appropriate for the PDF dates. Save and restore the catcode rather than fiddle around with scoping.

```
  \edef\@texosquery@restore@D{%
    \noexpand\catcode`\noexpand\D=\the\catcode`\D\relax}%
  \catcode`\D=12\relax
  \TeXOSQuery{#1}{\string-n}%
  \@texosquery@restore@D
}
```

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

\texosquerystripquotes

```
\def\texosquerystripquotes#1{%
  \@texosquery@stripquotes#1\@mid@texosquery@stripquotes
    "\relax"\relax\@end@texosquery@stripquotes
}
\def\@texosquery@stripquotes#1"#2"{%
  \@@texosquery@stripquotes#1#2%
}
\def\@@texosquery@stripquotes#1\@mid@texosquery@stripquotes#2\@end@texosquery@stripquotes{%
  #1%
}
```

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

```
\def\TeXOSQueryFileDate#1#2{%
  \edef\@texosquery@restore@D{%
    \noexpand\catcode`\noexpand\D=\the\catcode`\D\relax}%
  \catcode`\D=12\relax
  \TeXOSQuery{#1}{\string-d \string'\texosquerystripquotes{#2}\string'}%
  \@texosquery@restore@D
}
```

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

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

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

```
\def\TeXOSQueryFileList#1#2#3{\TeXOSQuery{#1}{%
 \string-i \string'#2\string'
 \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!*

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

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

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

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

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

`\TeXOSQueryDirName` Get the canonical path of the directory containing the file given in the second argument.

```
\def\TeXOSQueryDirName#1#2{\TeXOSQuery{#1}{\string-e
 \string'\texosquerystripquotes{#2}\string'}}
```

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

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

3.2 LaTeX Code

This is just a simple wrapper for `texosquery.tex` so that it can be loaded using LaTeX's standard `\usepackage` method. Identify package:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{texosquery}[2016/07/14 v1.1 (NLCT)]
```

Load `texosquery.tex`:

```
\input{texosquery}
```

That's it!

Change History