texosquery: query OS information from TeX

Nicola L. C. Talbot   Paulo Cereda

dickimaw-books.com

2020-02-04 (v1.7)

Abstract

The texosquery bundle provides the `texosquery.jar` application (and variations `texosquery-jre8.jar` and `texosquery-jre5.jar`) This is a cross-platform Java application to query certain operating system (OS) and locale information. The application is specifically designed for use within TeX's shell escape mechanism, through the `\TeXOSQuery` command provided by the texosquery package (`texosquery.tex` and `texosquery.sty`).

The `\TeXOSQuery` command performs more than a simple piped input as it first changes category codes of various problematic characters and locally defines some short control sequences that are used in the application's result. These commands aren't defined outside of `\TeXOSQuery`, so a direct piped input may cause undefined control sequences. If you really want to use this direct method rather than using `\TeXOSQuery`, then you will need to run `texosquery` in backward compatibility mode 0 or 1 (using `--compatible 1`). The first two versions of `texosquery` didn't use those short commands.

Important Note: You will need TeX's piped shell escape enabled and you will also need the Java Runtime Environment (JRE) installed. MiKTeX users will need to add `--enable-pipes` to TeX's command line options. (Alternatively, use `\TeXOSQueryFromFile` if you can't use the piped shell escape.) The configuration file `texosquery.cfg` should be edited to reflect your system set-up before use.

There are three variations of the `texosquery` application provided:

- `texosquery.jar`: requires at least Java 7, has medium locale support, obeys `openin_any` but has additional restrictions imposed for security reasons (no listings outside the current working directory path); Note that Java 7 has reached its end of life and is now deprecated.

- `texosquery-jre8.jar`: requires at least Java 8, has best locale support, is on TeX Live's restricted list, obeys `openin_any` but has additional restrictions imposed for security reasons (no listings outside the current working directory path);

- `texosquery-jre5.jar`: requires at least Java 5, has poor locale support (language scripts not recognised), doesn't have the walk action, obeys `openin_any` but doesn't

have the extra restrictions of the Java 7 and 8 versions for the listing functions. Note that Java 5 and 6 are deprecated. Old deprecated versions are considered a security risk.

The default is now `texosquery-jre8.jar`. If you have an older version of Java you will need to edit the `texosquery.cfg` file (although, if possible, you should consider upgrading your Java version for security reasons). Throughout this document `texosquery` is used to reference the application, regardless which of these three jar files you've chosen to use. See section 1.1 for further details.

The aim of the original version of `texosquery` was to provide a way of accessing the operating system's locale information. Version 1.3 of the `tracklang` package provides `\TrackLangQueryEnv` which uses `kpsewhich` to query the appropriate locale environment variable (such as `$LANG` or `LC_ALL`). Unfortunately this doesn't work under Windows as the locale information there is stored in the registry. The Lua `os.setlocale(nil)` function can simply return `C` or `POSIX`, which isn't helpful from `tracklang`'s point of view. Although Java has its drawbacks, it's one of the most ubiquitous platform-independent methods to obtain this information. Since it seemed overkill to write a Java application that simply returned the locale, I decided to add a few extra functions that might be of use, but accessing locale information was, and still is, the primary purpose of this application.

Although the POSIX environment variables, such as `$LC_ALL`, are easy to read with `kpsewhich`, these days the IETF BCP 47 language tag is the more appropriate way of identifying a locale, so version 1.2 has added the `--bcp47` function to support this. The `tracklang` package has similarly added `\TrackLanguageTag{⟨IETF tag⟩}`.

The `--numeric`, `--locale-data`, `--date-time` and `--time-zones` options are designed to interface with the tex-locale package, so although texosquery and tex-locale are distributed separately, version 1.2 of `texosquery` was developed alongside version 1.0 of the tex-locale package. The aim of the tex-locale package is to use both `tracklang` and `texosquery` to automatically set up the document language. For example, in the following LaTeX document

```
\documentclass{article}
\usepackage{tex-locale}
\begin{document}
Language: \CurrentLocaleLanguageNativeName.
Region: \CurrentLocaleRegionNativeName.
Today: \CurrentLocaleDate. (Compare with \today.)
Time: \CurrentLocaleTime.
Currency Symbol: \CurrentLocaleCurrency
Integer:
\texosqueryfmtnumber{\CurrentLocaleIntegerPattern}{123456}{0}{0}
Decimal:
\texosqueryfmtnumber{\CurrentLocaleDecimalPattern}{123456}{78}{0}
Percentage:
\texosqueryfmtnumber{\CurrentLocalePercentPattern}{0}{65}{0}
Currency:
\texosqueryfmtnumber{\CurrentLocaleCurrencyPattern}{1234567}{0}{0}
\end{document}
```

the tex-locale package will automatically:

- load the textcomp package for currency symbols (package option `symbols=fontawesome` will use fontawesome instead);

- if X∄LATEX or LuaLATEX:

    - load fontspec (unless option `fontspec=false` is used);
    - load polyglossia and use `\setmainlanguage` with options that can be determined from the language tag (use package option `support=babel` to use babel regardless of the LATEX format);

    otherwise:

    - load inputenc (default file encoding obtained from `texosquery`'s `--codeset-lcs` action);
    - load fontenc (font encoding obtained using tracklang to query the language script);
    - load babel with the appropriate language label (use `support=none` to prevent this);

- load datetime2 with the `useregional=text` option (use `datetime2=false` to prevent this).

The generic `locale.tex` code doesn't load the above packages, but can still obtain information about the locale:

```
\input locale

Language: \CurrentLocaleLanguageNativeName.
Region: \CurrentLocaleRegionNativeName.
Today: \CurrentLocaleDate. (Compare with \today.)
Time: \CurrentLocaleTime.
Currency Symbol: \CurrentLocaleCurrency
Integer:
\texosqueryfmtnumber{\CurrentLocaleIntegerPattern}{123456}{0}{0}
Decimal:
\texosqueryfmtnumber{\CurrentLocaleDecimalPattern}{123456}{78}{0}
Percentage:
\texosqueryfmtnumber{\CurrentLocalePercentPattern}{0}{65}{0}
Currency:
\texosqueryfmtnumber{\CurrentLocaleCurrencyPattern}{1234567}{0}{0}
\bye
```

So that's the reasoning behind the new v1.2 actions.

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. This information can 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. It also uses control sequence markup to indicate whether characters should be interpreted literally (such as in file names) or if they should obey their current category code (such as punctuation occurring in textual information) or if they should be interpreted in some other way (such as pattern markup). This markup is expanded by `\TeXOSQuery` when it performs the piped shell escape.

> **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 (through `\TeXOSQuery` defined in `texosquery.tex`) each action (indicated by a command line switch) will display the result on a single line. For multiple results, each line is grouped. A blank line (or empty group) will be displayed if the information isn't available or is prohibited. A forward slash (`\fslh`) is always used as a directory divider, regardless of the operating system, so the result can be used, for example, in `\input` or `\includegraphics`.

For example, I have a 64-bit Linux operating system installed on my computer, so I could use `uname` in a bash terminal:

```
uname -o -r
```

which (for me) produces:

```
4.1.13-100.fc21.x86_64 GNU/Linux
```

I could also run `texosquery` directly from the bash terminal:

```
texosquery -o -r
```

which produces the rather more cryptic:

```
{Linux}
{4\fdot 1\fdot 13\fhyn 100\fdot fc21\fdot x86\fusc 64}
```

However `texosquery` isn't intended for this direct use. It's intended for use with `\TeXOSQuery` provided by `texosquery.tex`. Here's a plain TeX document:

```
\input texosquery
\TeXOSQuery{\result}{-o -r}
\def\parseresult#1#2{OS Name: {\tt #1}. OS Version: {\tt #2}.}
\ifx\result\empty
 Query failed!
\else
 \expandafter\parseresult\result
\fi
\bye
```

The markup commands, such as \fusc, are now converted to literal characters with category code 12 ("other"), so the underscore isn't a problem. This document is now also platform independent (as long as texosquery and a recent version of the JRE are installed). Unlike uname, texosquery also obeys the order of the command line switches, which makes it easier to define the helper command (\parseresult in the above) that processes the result.

1.1 Installation and Setup

Installation is best done through your TeX package manager. However if for some reason you need to install this package manually the instructions are below. If you install through your package manager, Windows users will probably find that the .jar files have been converted to .exe (with the .bat files omitted) and Unix-like users may find that the bash scripts are missing the .sh extension (these are actually symbolic links to the distributed .sh files). See section 1.2 to test that the package has been successfully installed.

Even if you use your TeX distribution's package manager to install this package, you may still need to edit the texosquery.cfg file (see step 3 below). It's best to copy this file to your ⟨*TEXMFHOME*⟩ or ⟨*TEXMFLOCAL*⟩ tree to avoid losing your changes when the package is updated.

You can find the correct value of ⟨*TEXMFHOME*⟩ using

```
kpsewhich -var-value=TEXMFHOME
```

Similarly for ⟨*TEXMFLOCAL*⟩.

You can find where the package manager has put texosquery.cfg using

```
kpsewhich texosquery.cfg
```

This bundle contains the following files:

- texosquery.dtx

  The DTX file contains the source code for this document, and also the files:

  - texosquery.tex (generic TeX code)
  - texosquery.sty (LaTeX package wrapper)
  - texosquery.cfg (configuration file)

  The bash scripts (which will need the extensions removed):

  - texosquery-jre8.sh
  - texosquery.sh
  - texosquery-jre5.sh

  Windows batch files (which will need the extensions changed to .bat)

  - texosquery-jre8.batch
  - texosquery.batch

- texosquery-jre5.batch

- `texosquery.ins` The driver file used to extract all the above files contained in `texosquery.dtx`.

- The three different versions of the `texosquery` application: `texosquery-jre8.jar`, `texosquery.jar` and `texosquery-jre5.jar`. The source code for these is contained in the `java` sub-directory.

- `texosquery.pdf` This PDF document.

- `README.md` The README file in markdown format.

- `CHANGES` Lists major changes for each version.

To install manually (⟨*TEXMF*⟩ indicates the TEXMF directory):

1. Run

   ```
   tex texosquery.ins
   ```

   to extract the `.tex`, `.sty`, `.cfg`, `.sh` and `.batch` files.

   Windows Change the extension of the `.batch` files to `.bat` (TeX on Windows prohibits the creation of `.bat` files). Move the `.bat` files to somewhere on your system's path. (You may omit the `.bat` files you don't need.) The `.sh` files may be deleted.

   Unix-like Make the `.sh` files executable:

   ```
   chmod u+x texosquery*.sh
   ```

   Move the `.sh` files to somewhere on your path *without* the `.sh` extension. (If the `.sh` extension is retained, you will have to edit the `texosquery.cfg` file to include it.) For example (if `~/bin` is included in `$PATH`):

   ```
   mv texosquery-jre8.sh ~/bin/texosquery-jre8
   ```

   (You may omit the `.sh` files you don't need.) The `.batch` files may be deleted.

2. Move `texosquery.tex` to ⟨*TEXMF*⟩`/tex/generic/texosquery/`

3. Edit `texosquery.cfg` so that `\TeXOSInvokerName` is defined to the application of your choice. For example, if you have Java 8 installed:

   ```
   \def\TeXOSInvokerName{texosquery-jre8}
   ```

   Or if you only have Java 5 or 6 installed:

   ```
   \def\TeXOSInvokerName{texosquery-jre5}
   ```

You can find out your Java version by running the following in your command prompt or terminal:

```
java -version
```

If the version number starts with `1.8` then you have Java 8 installed, if it starts with `1.7` then you have Java 7, etc.

4. Move `texosquery.cfg` to ⟨*TEXMF*⟩`/tex/generic/texosquery/`

5. Move `texosquery.sty` to ⟨*TEXMF*⟩`/tex/latex/texosquery/`

6. Move the `.jar` files to ⟨*TEXMF*⟩`/scripts/texosquery/`

1.2   Installation Test

To test the installation:

1. In the command prompt or terminal do:

```
texosquery -b
```

(Replace `texosquery` with the command that matches the value of `\TeXOSInvokerName` in the `texosquery.cfg` file described in section 1.1.) The above command should display the system's default locale. For me, this simply displays the line:

```
en-GB
```

If you get an `Unknown option '-b'` error, then your OS is picking up an old version of `texosquery`. Check the version number with the `-v` switch.

```
texosquery -v
```

If you get a "command not found" or "bad command or file name" error, then recheck the installation steps in section 1.1 and make sure that the executable file has been placed on your system's path.

If this test is successful, try the next step.

2. Create the following plain TeX document called `test.tex`:

```
\input texosquery
\TeXOSQuery{\result}{-b}\result
\bye
```

and compile using:

```
pdftex --shell-escape test
```

9

Alternatively, create the follow LATEX document called `test.tex`:

```
\documentclass{article}
\usepackage{texosquery}
\begin{document}
\TeXOSQuery{\result}{-b}\result
\end{document}
```

and compile using:

```
pdflatex --shell-escape test
```

In both cases, the resulting PDF file `test.pdf` should show the default locale. If not check the transcript `test.log` which should include something like (`|texosquery -b`) or (`|texosquery-jre8 -b`) etc. If it simply has the line:

```
TeXOSQuery: texosquery -b
```

(or similar) then the dry run mode was on, which means the shell escape wasn't used. Check that the `--shell-escape` switch was used when calling `pdftex` or `pdflatex`.

TEX Live 2017 has added `texosquery-jre8` to the restricted list, but you need to modify the configuration file to take advantage of this. Make sure that the line

```
\TeXOSQueryAllowRestricted
```

hasn't been commented out in the `texosquery.cfg` file and try the above example documents in restricted mode.

1.3   Accessing file information

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). As from version 1.2, `texosquery` honours the `openin_any` attribute set in the `texmf.cnf` configuration file. This value is fetched using

```
kpsewhich -var-value=openin_any
```

(You can find the configuration files using `kpsewhich -a texmf.cnf`) For example, suppose the file `/tmp/.test` exists. If the `openin_any` attribute is set to "a" (any file), then (assuming the operating system allows read-access to that file) the `texosquery` file-reading operations will be permitted. For example

```
texosquery --pdfdate /tmp/.test
```

will return the file modification date in PDF date-time format. However, if `openin_any` is set to "`r`" (restricted), the read access will be denied because the file is considered hidden so an empty result is returned. Similarly, if `openin_any` is set to "`p`" (paranoid), the read access will be denied again because the file is hidden but also because the file has an absolute path that isn't under `$TEXMFOUTPUT` (assuming that environment variable hasn't been set to `/tmp`).

1.4   MiKTeX

MiKTeX doesn't support the `openin_any` variable so, if this is unset, `texosquery` will fallback on "`a`".

Note that MiKTeX disables piped input by default for security reasons. Since `\TeXOSQuery` relies on piped input, you'll need to enable it with `--enable-pipes` when you run TeX.

1.5   Restricted Mode

TeX Live 2017 now has `texosquery-jre8` on the list of trusted applications that may be run in restricted mode. In order to make use of this, you need to set up your configuration file to use both `texosquery-jre8` and to automatically switch off the dry run mode if the restricted shell escape is detected. However, note that the restricted mode has limitations on the characters allowed in the shell escape for security reasons. This means that arguments (such as file names) can't be quoted in restricted mode and therefore arguments that contain spaces can't be delimited and will cause problems. Either avoid spaces in file names or use the unrestricted mode.

1.6   Return Values

The return values may include literal text where special characters need to have their category code changed to 12 (for example, file names) but the return values may also include TeX code that needs to be processed by TeX, either during the shell escape or deferred for later (such as date-time or numeric patterns). This means that the result from the shell escape can't be automatically detokenized.

Therefore, as from version 1.2, the return values include short control sequences that are locally defined by `\TeXOSQuery` and so are only valid within that command's scope. For example, `\fcln` expands to a colon (`:`) with category code 12 whereas `\tcln` expands to a colon according to its current meaning. Note that this has changed from earlier versions which simply returned the actual characters, which may or may not have had the category code set to 12 at the start of `\TeXOSQuery`. To reproduce the original behaviour, use the compatibility mode (`--compatible`) with the level set to 0 or 1. For the full list of shortcut commands, see the definition of `\@texosquery@enableshortcs`.

If the operating system uses a backslash `\` as a directory divider, returned path names will always replace this with a forward slash `/` (which is then converted to `\fslh`, as described above). This is designed to make the result as compatible as possible with TeX file commands (such as `\input`) which require a forward slash. However, for some

systems, further modification may need to be done. For example, with TeX on Cygwin, the Windows path name `c:\cygwin64\usr\local\texlive` may correspond to the TeX path name `/usr/local/texlive`. This path is returned by `texosquery` as `c:/cygwin64/usr/local/texlive` (`\fcln` and `\fslh` replaced for clarity). You can use `--strip-path-prefix` to strip the leading `c:/cygwin64` for convenience (see section 1.8).

The output produced by the `texosquery` application will be returned using the system's default file encoding. (For example, UTF-8.) You will need to ensure that your TeX document uses the same encoding if you want to typeset any of the results that may contain non-ASCII characters. You can determine the default encoding with `texosquery -C`, which is formatted to match the options used by the inputenc package. (For example, `uft8` for UTF-8.)

To test the file encoding rerun the plain TeX or LaTeX test file in section 1.2 with `-N` instead of `-b`. Most currency symbols are outside the ASCII set, so this should return a non-ASCII character. If you happen to have $ as your currency, then try `en-GB` or `en-IE` which have £ and €, respectively. LaTeX users may need to load inputenc and fontenc. XeLaTeX and LuaLaTeX users may need to load fontspec.

As from version 1.6, you can now override the default encoding using the `--encoding` option. This can be set on a document basis, for example:

```
\usepackage[utf8]{inputenc}
\TeXOSQuery{\result}{--encoding UTF-8 -N}
```

or for all documents by editing the `texosquery.cfg` file to include the option in the invoker command.

1.7 Locales

The options that have a locale identifier as an argument need the identifier formatted as a *regular* IETF BCP 47 language tag that uses hyphens as separators. POSIX style locales (with underscores replaced by hyphens, for example `fr-BE.utf8@euro`) are only used as a return value in the `--locale` and `--locale-lcs` options.

Not all locales are supported by Java. For example, Irish is supported but Scottish and Welsh aren't supported by the JRE. The Unicode Consortium's Common Locale Data Repository (CLDR) can be accessed with Java 8, but the CLDR isn't enabled by default. It can be turned on using the system property `java.locale.providers`, which may provide additional support. For example, although Welsh isn't supported by the JRE, it is supported with the CLDR, so both Java 8 and the CLDR locale provider are required for that language. The proposed Java 9 should have the CLDR enabled by default.

The bash script `texosquery-jre8` automatically sets `java.locale.providers` to `CLDR,JRE`. Alternatively, the `JAVA_TOOL_OPTIONS` environment variable can be set to

```
-Djava.locale.providers=CLDR,JRE
```

which will enable it for all installed Java applications. If you need to set multiple options, these can be combined in the value of `JAVA_TOOL_OPTIONS`. For example

```
-Djava.locale.providers=CLDR,JRE -Dfile.encoding=UTF-8
```

You may find that the results are different depending on the data provider. For example with `java.locale.providers` set to `JRE,CLDR` then

```
texosquery -D en-GB
```

displays the long date in the form "06 November 2016" and the medium date in the form "06-Nov-2016", but with the ordering reversed to `CLDR,JRE` (so that the CLDR is queried first) then the long date is now in the form "6 November 2016" and the medium date is in the form "6 Nov 2016".

Note that `texosquery` can only access locale information provided by Java. For example, Java currently doesn't provide any methods to access telephone codes.

1.8   Command line invocation

The syntax for the command line invocation of `texosquery` is:

`texosquery` [⟨*options*⟩] ⟨*action*⟩ . . .

The syntax for `texosquery-jre8` is exactly the same except for the application name:

`texosquery-jre8` [⟨*options*⟩] ⟨*action*⟩ . . .

Similarly for `texosquery-jre5`. (Bash users may need the `.sh` extension if it wasn't removed from the script name during the installation setup, but it's best to remove it.) Available actions are described below. At least one action is required.

Available options (must come before actions):

`-h` **or** `--help` **or** `-help` Displays help message and exits.

`-v` **or** `--version` **or** `-version` Displays version information and exits.

`--nodebug` No debugging information. Only command line syntax errors are written to STDERR. (Default.)

`--debug` [⟨***n***⟩] **or** `-debug` [⟨***n***⟩] Set the debugging level, where ⟨*n*⟩ is a non-negative integer. If ⟨*n*⟩ is omitted, 3 is assumed. If ⟨*n*⟩ is 0, then debugging information is suppressed (equivalent to `--nodebug`). If ⟨*n*⟩ ≥ 1, error messages are written to STDERR. If ⟨*n*⟩ ≥ 2, any exceptions encountered will additionally write the stack trace to STDERR. If ⟨*n*⟩ ≥ 3 non-error informational messages are included.

`--compatible` ⟨***n***⟩ **or** `-compat` ⟨***n***⟩ Set the compatibility mode. The argument should be either a non-negative integer (0 for version 1.0, 1 for version 1.1, 2 for version 1.2 onwards) or the keyword `latest` to indicate the latest version (default). Note that the compatibility mode only affects the available actions and the display style of the result, and does not change security features. For example, the check for the `openin_any` setting was only introduced to version 1.2, but this is still checked even if the compatibility mode is set to 0 or 1.

`--encoding` ⟨***charset***⟩ **or** `-enc` ⟨***charset***⟩ (New to version 1.6.) This option may be used to override the default file encoding. For example, if your native file encoding is CP1250, but your document uses UTF-8 then you'll need to use `--encoding UTF-8`.

`--default-encoding` **or** `-defenc` (New to version 1.6.) This option is the default, but may be used to cancel the effect of `--encoding`. (For example, if you have added `--encoding` to the invoker command in the `texosquery.cfg` file, but you want to cancel it in a particular document.)

The following options (introduced in v1.5) allow returned paths or URIs to be altered (for example, if a mount point needs to be removed). Note that these options only affect *returned* paths, not path names provided in the command line. The substitution is performed after any backslash directory dividers \ are changed to forward slashes / (if applicable) but before non-alphanumerics are changed to control sequences (such as `\fcln` or `\fslh`). If there's no match, the path is returned without alteration.

`--strip-path-prefix` ⟨***prefix***⟩ **or** `-sp` ⟨***prefix***⟩ Strip ⟨*prefix*⟩ from the start of returned path names (but not URIs, see below). For example:

```
texosquery --strip-path-prefix c:/cygwin64 -p article.cls
```

This will strip the `c:/cygwin64` mount point. Note that ⟨*prefix*⟩ isn't an expression. For a regular expression, use `--replace-path`. (This option can't be used with `--replace-path`.)

`--nostrip-path-prefix` Cancels the effect of `--strip-path-prefix`.

`--replace-path` ⟨***regex***⟩ ⟨***replacement***⟩ **or** `-rp` ⟨***regex***⟩ ⟨***replacement***⟩ This replaces the *first* occurrence of the regular expression ⟨*regex*⟩ in any returned path (but not URI) and replaces it with ⟨*replacement*⟩.

`--noreplace-path` Cancels the effect of the `--replace-path`.

`--strip-uri-prefix` ⟨***prefix***⟩ **or** `-su` ⟨***prefix***⟩ Replace initial `file:/`⟨*prefix*⟩ with just `file:/` from returned URIs. Note that ⟨*prefix*⟩ isn't an expression. For a regular expression, use `--replace-uri`. (This option can't be used with `--replace-uri`.)

`--nostrip-uri-prefix` Cancels the effect of `--strip-uri-prefix`.

`--replace-uri` ⟨***regex***⟩ ⟨***replacement***⟩ **or** `-ru` ⟨***regex***⟩ ⟨***replacement***⟩ This replaces the *first* occurrence of the regular expression ⟨*regex*⟩ in any returned URI and replaces it with ⟨*replacement*⟩. For example:

```
texosquery -ru file:/C: file://cgwin64/c -u filename
```

`--noreplace-uri` Cancels the effect of the `--replace-uri`.

If multiple actions 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\fhyn GB\fdot utf8
```

(which expands to `en-GB.utf8` when used with `\TeXOSQuery`) whereas

```
texosquery -l -o
```

produces:

```
{en\fhyn GB\fdot utf8}
{Linux}
```

To reproduce the v1.0 display use `--compatible 0`. (This will also explicitly use the punctuation characters rather than replacing them with the control sequence markup, such as `\fhyn` or `\fdot`.)

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

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

produces:

```
{en\fhyn GB\fdot utf8}
{}
```

whereas

```
texosquery -d nofile
```

just displays an empty line.

If you're puzzled as to why an empty line has been returned, try rerunning the command with `--debug` for further information. Available actions are listed below.

Action `-b` **or** `--bcp47`

(New to version 1.2.) This action displays the BCP 47 language tag. For example, my locale is `en-GB` (English in the United Kingdom), so

```
texosquery -b
```

Simply returns:

```
en\fhyn GB
```

(which expands to `en-GB` with `\TeXOSQuery`) whereas a user whose default locale is set to Swiss German with the new orthography would get:

```
de\fhyn CH\fhyn 1996
```

(which expands to `de-CH-1996` with `\TeXOSQuery`).

Action -L or --locale

This action displays the locale information in the POSIX 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. As above, the punctuation characters will actually be returned using the control sequences \fhyn (hyphen), \fdot (dot) and \fatc (at).

If the --encoding option is used, the codeset will reflect that value.

Action -l or --locale-lcs

This action is similar to --locale, but the codeset 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. As above, the punctuation characters will actually be returned using the control sequences \fhyn (hyphen), \fdot (dot) and \fatc (at). Again, if the --encoding option is used, the codeset will reflect that value.

Action -cs or --codeset

(New to version 1.6.) This action returns just the codeset. For example, my default file encoding is UTF-8, so

```
texosquery --codeset
```

returns

```
UTF-8
```

whereas

```
texosquery --encoding ISO-8859-1 --codeset
```

returns

```
ISO-8859-1
```

(In both the above examples, the hyphen character is actually returned as the control sequence \fhyn.)

Action -C or --codeset-lcs

(New to version 1.2.) This action returns just the codeset converted to lower case with hyphens stripped. For example, my default file encoding is UTF-8, so

```
texosquery -C
```

returns

```
utf8
```

whereas

```
texosquery --encoding ISO-8859-1 -C
```

returns

```
iso88591
```

Action `-o` **or** `--osname`

This action displays the operating system name. For example, for me this produces:

```
Linux
```

Action `-r` **or** `--osversion`

This action displays the operating system version. For example, for me this produces:

```
4\fdot 1\fdot 13\fhyn 100\fdot fc21\fdot x86\fusc 64
```

(which expands to `4.1.13-100.fc21.x86_64` when used with `\TeXOSQuery`).

Action `-a` **or** `--osarch`

This action displays the operating system architecture. For example, for me this produces:

```
amd64
```

Action `-M` **or** `--date-time`

(New to version 1.2.) This action displays all the current date time data in a format suitable for use in `\texosqueryfmtdatetime`. (See section 3.1.4.)

Action `-Z` [⟨*locale*⟩] **or** `--time-zones` [⟨*locale*⟩]

(New to version 1.2.) This action displays all of the time zone mappings for the given locale (or the default if ⟨*locale*⟩ is omitted) in the format

{{⟨*id₁*⟩}{⟨*short name*⟩}{⟨*long name*⟩}{⟨*dst short name*⟩}{⟨*dst long name*⟩}}...{{⟨*idₙ*⟩}{⟨*short name*⟩}{⟨*long name*⟩}{⟨*dst short name*⟩}{⟨*dst long name*⟩}}

The ⟨*id*⟩ is the unique label used by Java to identify the time zone (such as `Europe/London`) as used in the time zone information returned by `-M` (`--date-time`).

Action `-n` or `--pdfnow`

This action displays the current date and time in PDF format. For example

```
\pdfd \fcln 20160704131006\fpls 01\fapo 00\fapo
```

This uses the shorthand tags `\pdfd`, `\fcln`, `\fpls` and `\fapo` that are locally redefined by `\TeXOSQuery` to produce a D, a colon (`:`), a plus sign (`+`) and an apostrophe (`'`) with the category code set to 12 to make it consistent with `\pdfcreationdate`. This also allows for situations where the punctuation characters have been made active (for example, through babel).

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).

Note that versions 1.0 and 1.1 didn't use `\pdfd` etc but simply used the actual characters. For example:

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

If you want to reproduce this format, use `--compatible` with the level set to 0 or 1.

Action `-d` ⟨*file*⟩ or `--pdfdate` ⟨*file*⟩

This action 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.

As with `--pdfnow` this now uses `\pdfd` etc which are converted by `\TeXOSQuery` to characters with the category code set to 12.

This action obeys the `openin_any` setting, so if access to ⟨*file*⟩ is forbidden by this setting, the result will be empty.

Action `-s` ⟨*file*⟩ or `--filesize` ⟨*file*⟩

This action displays the size in bytes of the given file or an empty string 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.

This action obeys the `openin_any` setting, so if access to ⟨*file*⟩ is forbidden by this setting, the result will be empty.

Action `-c` or `--cwd`

This action displays the current working directory. This obeys the `openin_any` setting, so this action will return an empty string if this file information is forbidden by that setting.

Action `-m` or `--userhome`

This action displays the user's home directory. This obeys the `openin_any` setting, so this action will return an empty string if this file information is forbidden by that setting.

Action `-t` **or** `--tmpdir`

This action displays the temporary directory. This obeys the `openin_any` setting, so this action will return an empty string if this file information is forbidden by that setting.

Action `-i` ⟨*sep*⟩ ⟨*dir*⟩ [⟨*sort*⟩] **or** `--list` ⟨*sep*⟩ ⟨*dir*⟩ [⟨*sort*⟩]

This action lists 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.

> **Important Note:** As from v1.2, new restrictions have been placed on the value of ⟨*dir*⟩ for security reasons. For all three applications, `texosquery-jre8`, `texosquery` and `texosquery-jre5`, the openin_any setting is checked. If read access to ⟨*dir*⟩ is forbidden by the `openin_any` setting, then this action returns an empty string. *Additionally*, regardless of `openin_any`, the more restrictive applications, `texosquery-jre8` and `texosquery`, prohibit a value of ⟨*dir*⟩ that's outside the current working directory path (e.g. `..`) or that has no parent directory (e.g. `/`). Both `texosquery-jre8` and `texosquery` check the *canonical path* of ⟨*dir*⟩, so if ⟨*dir*⟩ is a symbolic link, the target path is checked.
>
> This is a security feature to prevent any malicious code that might try to recursively list the contents of the entire filing system, which would hog resources, or that might try to discover files outside the current working directory. An exception is made for `texosquery-jre5` since that application is already considered insecure (due to Java 5 and 6 being long deprecated), so if you really need ⟨*dir*⟩ as, say `..` (the parent directory) or `/` (the root directory), you can use `texosquery-jre5` (by redefining `\TeXOSInvokerName` before using `\TeXOSQuery`) although this isn't recommended. It will still obey the `openin_any` setting, so the listing still won't work with `texosquery-jre5` if the `openin_any` setting is set to `p` (paranoid).

As from version 1.2, there is now an optional argument ⟨*sort*⟩, which indicates how the returned list should be sorted. If omitted `default` is assumed. Available values of ⟨*sort*⟩:

`default` Use the default order. This is typically in alphabetical order, but depends on the operating system or JRE.

`date-ascending` Order by file modified date from oldest to newest. This option has synonyms `date` and `date-asc`.

`date-descending` Order by file modified date from newest to oldest. You may use the shorter `date-des` value instead.

`size-ascending` Order by file size from smallest to largest. This option has synonyms `size` and `size-asc`.

size-descending Order by file size from largest to smallest. You may use the shorter `size-des` value instead.

name-ascending Order by file name (case-sensitive) alphabetically. This option has synonyms `name` and `name-asc`.

name-descending Order by file name (case-sensitive) in reverse alphabetic order. You may use the shorter `name-des` value instead.

iname-ascending Order by file name (case-insensitive) alphabetically. This option has synonyms `iname` and `iname-asc`.

iname-descending Order by file name (case-insensitive) in reverse alphabetic order. You may use the shorter `iname-des` value instead.

ext-ascending Order by file extension (case-sensitive) alphabetically. If files have the same extension, they are ordered by name. This option has synonyms `ext` and `ext-asc`.

ext-descending Order by file extension (case-sensitive) in reverse alphabetic order. If files have the same extension, they are ordered by name (reverse alphabetic order). You may use the shorter `ext-des` value instead.

This action obeys the `openin_any` setting for all the listed files as well as for the directory ⟨*dir*⟩, so if access to a file in the directory is forbidden, the file will be omitted from the list. (This action is equivalent to the following with ⟨*regex*⟩ set to `.*` to match all files.)

If you want to excluded hidden dot files (where they aren't automatically excluded by `openin_any`), use the `--filterlist` action described below with ⟨*regex*⟩ set to `[^\.].*`. (Remember that you'll need to use `\string` when using the shell escape, as noted below.)

> **Important Note:** Unlike most of the return values the ⟨*sep*⟩ part here isn't escaped, so take care if ⟨*sep*⟩ contains any commands. For example, if you want to use `\\` as the separator, you'll need to use `\string\noexpand\string\\` in the ⟨*sep*⟩ part within `\TeXOSQuery`.

For example:

`\TeXOSQueryFileList{\result}{\string\noexpand\string\\}{.}`

calls (through the shell escape):

`texosquery -i '\noexpand\\' '.'`

(the two `\string` commands have detokenized their arguments) so `texosquery` uses `\noexpand\\` as the separator in the returned list, but this list is expanded as it's read in. However `\noexpand` prevents the `\\` from being expanded, so the separator becomes just `\\` which may be (re)defined before the resulting list is processed.

Note that `\TeXOSQueryFileList` automatically adds the single quotes around the arguments. If `\TeXOSQuery` is used explicitly, these quotes would need to be added as appropriate.

Action -id ⟨*sep*⟩ ⟨*dir*⟩ [⟨*sort*⟩] **or** --list-dir ⟨*sep*⟩ ⟨*dir*⟩ [⟨*sort*⟩]

This action is like --list but only includes sub-directories of ⟨*dir*⟩. The caveats and security notes for --list also apply here.

Action -ir ⟨*sep*⟩ ⟨*dir*⟩ [⟨*sort*⟩] **or** --list-regular ⟨*sep*⟩ ⟨*dir*⟩ [⟨*sort*⟩]

This action is like --list but only includes regular files. The caveats and security notes for --list also apply here.

Action -f ⟨*sep*⟩ ⟨*regex*⟩ ⟨*dir*⟩ [⟨*sort*⟩] **or** --filterlist ⟨*sep*⟩ ⟨*regex*⟩ ⟨*dir*⟩ [⟨*sort*⟩]

This action is like --list but only lists those files whose name matches the regular expression given in ⟨*regex*⟩. Note that since this uses Java's String.matches method this tests for a *complete* match on the file name (not including directory 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). Use .*foo.* to match all files that contain foo in the name (so foobar and barfoo will both match).

> **Important Note:** You can't have an empty regular expression. You can use the regular expression .* to match all files (which is what --list does).

As from version 1.2, this action now has an optional argument ⟨*sort*⟩, which indicates how to sort the returned list. The available values for ⟨*sort*⟩ are the same as for --list, described above.
The caveats and security notes for --list also apply here.

Action -fd ⟨*sep*⟩ ⟨*regex*⟩ ⟨*dir*⟩ [⟨*sort*⟩] **or** --filterlist-dir ⟨*sep*⟩ ⟨*regex*⟩ ⟨*dir*⟩ [⟨*sort*⟩]

This action is like --filterlist but only includes sub-directories of ⟨*dir*⟩.
The caveats and security notes for --list also apply here.

Action -fr ⟨*sep*⟩ ⟨*regex*⟩ ⟨*dir*⟩ [⟨*sort*⟩] **or** --filterlist-regular ⟨*sep*⟩ ⟨*regex*⟩ ⟨*dir*⟩ [⟨*sort*⟩]

This action is like --filterlist but only includes regular files.
The caveats and security notes for --list also apply here.

Action -w ⟨*sep*⟩ ⟨*regex*⟩ ⟨*dir*⟩ [⟨*sort*⟩] **or** --walk ⟨*sep*⟩ ⟨*regex*⟩ ⟨*dir*⟩ [⟨*sort*⟩]

(New to version 1.2. Not available with texosquery-jre5.)
This action starts from the directory ⟨*dir*⟩ *which must be on the current working directory's path* and returns a list separated by ⟨*sep*⟩ of all the regular files whose basename matches the regular expression ⟨*regex*⟩ (as for the filtered file listings described above), recursively descending sub-directories. Any files or sub-directories that are hidden, unreadable or symbolic links are skipped. The list is sorted according to ⟨*sort*⟩, which is as for the file listing actions described above. Note that ⟨*dir*⟩ is first converted to its canonical

path, so if ⟨*dir*⟩ is a symbolic link, the security check will test if the *target* path is on the current working directory path.

As with the above file listings, the separator ⟨*sep*⟩ isn't escaped so take care if ⟨*sep*⟩ contains any commands. The resulting list will consist of paths relative to ⟨*dir*⟩.

> **Important Note:** This action requires at least Java 7 so it's not available with `texosquery-jre5.jar`.

Action `-u` ⟨***file***⟩ **or** `--uri` ⟨***file***⟩

This action displays the URI of the given file or an empty string if the file doesn't exist or if the file permissions or the `openin_any` setting prohibit read access.

Action `-p` ⟨***file***⟩ **or** `--path` ⟨***file***⟩

This action displays the canonical path of the given file or an empty string if the file doesn't exist or if the file permissions or the `openin_any` setting prohibit this action.

Action `-e` ⟨***file***⟩ **or** `--dirname` ⟨***file***⟩

(New to v1.1.) This action displays the canonical path of the given file's parent (that is, the directory containing ⟨*file*⟩) or an empty string if the file doesn't exist or if the file permissions or the `openin_any` setting prohibits 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.

Action `-N` [⟨***language tag***⟩] **or** `--numeric` [⟨***language tag***⟩]

(New to v1.2.) This action displays:

{⟨*locale tag*⟩}{⟨*group sep*⟩}{⟨*decimal sep*⟩}{⟨*exp sep*⟩}{⟨*use group*⟩}{⟨*currency code*⟩}{⟨*regional currency code*⟩}{⟨*currency sym*⟩}{⟨*T̲EX currency*⟩}{⟨*currency sep*⟩}

for the ⟨*language tag*⟩ given in the optional argument. If omitted, the default locale is assumed. The returned values are:

- ⟨*tag*⟩ the language tag.

- ⟨*group sep*⟩ the numeric group separator.

- ⟨*decimal sep*⟩ the decimal separator.

- ⟨*exp sep*⟩ the exponent separator.

- ⟨*use group*⟩ `1` if the locale uses number grouping otherwise `0`.

- ⟨*currency code*⟩ the ISO 4217 currency code.

- ⟨*regional currency code*⟩ either the ISO 4217 currency code or an unofficial code. The only unofficial codes returned are: GGP (Guernsey pound), JEP (Jersey pound), IMP (Isle of Man pound), KID (Kiribati dollar) and TVD (Tuvaluan dollar).

- ⟨*currency sym*⟩ the currency symbol. (This may sometimes be the same as ⟨*currency code*⟩.) Non-ASCII characters will be marked up with \twrp (see below).

- ⟨*TEX currency*⟩ the currency symbol using TEX code provided by texosquery. This is obtained by substituting known Unicode currency symbols occurring in ⟨*currency sym*⟩ with \texosquerycurrency{⟨*xxx*⟩}, which expands to the control sequence given by the name texosquerycurrency⟨*xxx*⟩. These commands are defined in texosquery.tex. Since there are no generic TEX commands available for all these symbols (except $), these commands will need to be redefined as appropriate but are provided in the event that there's no UTF-8 support. There is a limited check for some known currency commands, such as \texteuro or \euro, but if an appropriate currency command can't be found, the \texosquerycurrency⟨*xxx*⟩ commands will be defined to simply the currency label (usually the same as the ⟨*xxx*⟩ part).

- ⟨*currency sep*⟩ the currency decimal separator.

The language tag should conform to IETF BCP 47. See http://docs.oracle.com/javase/8/docs/api/java/util/Locale.html for further details. If you are using texosquery-jre5, only the language, region and variant elements will be recognised since the language tag support was introduced in Java 7.

For example:

```
texosquery -N en-GB
```

produces

```
{en-GB}{,}{.}{E}{1}{GBP}{GBP}{\twrp{£}}{\texosquerycurrency{pound}}{.}
```

The \twrp command is used by texosquery to markup a non-ASCII character. This command is one of the shorthands only defined within \TeXOSQuery. In this case it's a shortcut for the command \texosquerynonasciiwrap. By default this just does its argument, but it may be redefined to perform some other action such as converting from one encoding to another.

In most cases the ⟨*regional currency code*⟩ will be the same as ⟨*currency*⟩ code. A few non-ambiguous unofficial codes are known by texosquery and may be used if the country code is recognised. For example,

```
texosquery -N en-IM
```

produces

```
{en-IM}{,}{.}{E}{1}{GBP}{IMP}{M\twrp{£}}{M\texosquerycurrency{pound}}{.}
```

If Java doesn't support the given locale, the currency code will appear as XXX with the symbol ¤ (generic currency sign).

> **Important Note:** This option and the following (--locale-data) are best used with XƎTEX or LuaTEX to deal with the non-ASCII characters. Make sure the file encoding used by Java matches the TEX file.

(See section 1.7 for the difference in locale providers.)

Action `-D` [⟨*language tag*⟩] **or** `--locale-data` [⟨*language tag*⟩]

(New to v1.2.) This action provides more extensive information than `--numeric`. The result has nested groups to assist parsing. Again the ⟨*language tag*⟩ may be omitted. For example,

```
texosquery --locale-data
```

For the default locale or

```
texosquery --locale-data en-GB
```

for the locale identified by `en-GB`. As with all the other actions, the result is written to STDOUT on a single line. Its overall length and the use of the shortcut commands used by `texosquery` to markup certain elements mean that it's not particularly human-readable, but it's designed to be easy for TeX to interpret. The information is returned in the following format:

{⟨*locale block*⟩}{⟨*current date block*⟩}{⟨*date pattern block*⟩}{⟨*current time block*⟩} {⟨*time pattern block*⟩}{⟨*current date time block*⟩}{⟨*date time pattern block*⟩}{⟨*days of the week block*⟩}{⟨*abbreviated dates of the week block*⟩}{⟨*month names block*⟩}{⟨*abbreviated month names block*⟩}{⟨*standalone days of the week block*⟩}{⟨*abbreviated standalone days of the week block*⟩}{⟨*standalone month names block*⟩}{⟨*abbreviated standalone month names block*⟩}{⟨*numeric block*⟩}{⟨*numeric patterns block*⟩}

There may seem to be some repetition here with the month and week day names, but with `texosquery-jre8`, the second set are the standalone version (for example, for a column header). In some languages, these may be different from the names used in the date format. Since this is new to Java 8, it's not supported in `texosquery.jar` or `texosquery-jre5.jar` and they simply reproduce the non-standalone names.

The information supplied with this option is quite complex, but it's used by the texlocale package to set up all the required information for each locale used in the document. Any non-ASCII characters are marked up with `\twrp`, which is locally defined by `\TeXOSQuery` to expand to `\texosquerynonasciiwrap`. This may be redefined to deal with the characters if necessary. For example, if the character needs to be converted from one encoding to another.

The blocks are:

⟨*locale block*⟩

    The locale information in the form:

    {⟨*tag*⟩}{⟨*language name*⟩}{⟨*locale language name*⟩}{⟨*region name*⟩}{⟨*locale region name*⟩}{⟨*variant name*⟩}{⟨*locale variant name*⟩}

    The ⟨*tag*⟩ is the language tag (the same format as `--bcp47`).

    The ⟨*language name*⟩ is the language name in the operating system's default locale.

    The ⟨*locale language name*⟩ is the language name in the locale's language.

For example, my locale is `en-GB`, so if I use

```
texosquery -D en-GB
```

then both ⟨*language name*⟩ and ⟨*locale language name*⟩ will be `English`, but if I use:

```
texosquery -D fr-GB
```

then ⟨*language name*⟩ will be `French` and the ⟨*locale language name*⟩ will be `fran\twrp{ç}ais` (note the non-ASCII character has been marked up). The locale tag `fr-GB` indicates that I'm writing in French but I'm in the United Kingdom (so the currency should be GBP).

The ⟨*region name*⟩ is the region's name in the operating system's default language.

The ⟨*locale region name*⟩ is the region's name in the locale's language.

So for me with `-D en-GB` I get `United\tspc Kingdom` for both ⟨*region name*⟩ and ⟨*locale region name*⟩. This illustrates another of the shorthand commands that `texosquery` uses that's only locally defined within `\TeXOSQuery`. In this case, `\tspc` just expands to a space. This is used to avoid accidentally discarding any intentional spaces that might follow a command name or any intentional consecutive spaces.

If, however, I use `-D fr-GB` I still get `United\tspc Kingdom` in ⟨*region name*⟩, but ⟨*locale region name*⟩ is now `Royaume-Uni`.

The ⟨*variant name*⟩ is the language's variant. For example, with `de-CH-1996` (Swiss German using the new orthography), the variant is 1996. There's no variant in `en-GB` so this value is empty for me.

The ⟨*locale variant name*⟩ is the variant in the locale's language. In the case of `de-CH-1996` this is still 1996.

⟨*current date block*⟩

This returns the current date in four different formats and also an integer that indicates the first day of the week in the given locale:

{⟨*full date*⟩}{⟨*long date*⟩}{⟨*medium date*⟩}{⟨*short date*⟩}{⟨*first day*⟩}

The actual date formats depend on the locale. For example, with `en-GB` the ⟨*full date*⟩ is (assuming today is 2016-11-08):

```
Tuesday,\tspc 8\tspc November\tspc 2016
```

(Tuesday, 8 November 2016). The ⟨*long date*⟩ is

```
08\tspc November\tspc 2016
```

(08 November 2016). The ⟨*medium date*⟩ is

```
08\thyn Nov\thyn 2016
```

(08-Nov-2016). The ⟨*short date*⟩ is

```
08\tslh 11\tslh 16
```

(08/11/2016). Note that there's a difference between using the CLDR locale data and the JRE data. If I'm using the Java 7 compatible `texosquery.jar` which only uses JRE locale data, then I get the above results, but if I use the bash script `texosquery-jre8` which sets `java.locale.providers` to CLDR,JRE then I get a slightly different result. The ⟨*long date*⟩ is

```
8\tspc November\tspc 2016
```

(8 November 2016) and the ⟨*medium date*⟩ is

```
8\tspc Nov\tspc 2016
```

(8 Nov 2016). The ⟨*medium date*⟩ may be numeric or may be an abbreviated form of ⟨*long date*⟩, depending on the language and the locale provider. Some languages aren't supported by JRE but are supported by CLDR. (Some aren't supported by either, but there's a chance that those languages will eventually be added to the CLDR.) For example, if I use `-D cy-GB` with the JRE as the locale provider I just get the `en-GB` dates, but if I use the CLDR provider I get Welsh dates.

Note that the proposed Java 9 should automatically use the CLDR as the locale provider, which is being increasingly adopted by applications as a common data repository.

The first day of the week index is zero-based starting with Monday. This is done in order to be compatible with pgfcalendar. For example, with `-D en-GB` ⟨*first day*⟩ is 0 (Monday), but with `pt-BR` ⟨*first day*⟩ is 6 (Sunday). The tex-locale package provides a way of converting the index to Monday=1 or Sunday=1 indexing.

⟨*date pattern block*⟩

The pattern used to format the full date, long date, medium date and short date. This is in the form:

{⟨*full pattern*⟩}{⟨*long pattern*⟩}{⟨*medium pattern*⟩}{⟨*short pattern*⟩}

Each pattern uses shorthand mark-up that's only locally defined within `\TeXOSQuery`. These short commands are expanded to longer commands provided by `texosquery.tex` to avoid name clashing with other packages. When used directly in the document text, these expand to reproduce the pattern.

For example, with `-D en-GB` I get the following pattern for the short date:

```
\patdtf{2}{d}\tslh \patdtf{2}{M}\tslh \patdtf{2}{y}
```

When parsed by \TeXOSQuery, this is internally converted to

`\texosquerydtf{2}{d}/\texosquerydtf{2}{M}/\texosquerydtf{2}{y}`

But default this simply expands to dd/MM/yy but may be used in the first argument of \texosqueryfmtdatetime. See sections 3.1.3 and 3.1.4 for further details.

⟨*current time block*⟩

The current time provided in various formats suitable to the given locale:

{⟨*full time*⟩}{⟨*long time*⟩}{⟨*medium time*⟩}{⟨*short time*⟩}

As with the current date, the actual format depends on the locale and the locale provider. For example, with en-GB I get:

`{15:59:41\tspc o\csq clock\tspc GMT}{15:59:41\tspc GMT}{15:59:41}{15:59}`

with the JRE. If I switch to CLDR (in the bash script texosquery-jre8) I get:

`{16:00:51\tspc Greenwich\tspc Mean\tspc Time}{16:00:51\tspc GMT}{16:00:51}{16:00}`

⟨*time pattern block*⟩

The pattern used to format the full time, long time, medium time and short time.

{⟨*full time pattern*⟩}{⟨*long time pattern*⟩}{⟨*medium time pattern*⟩}{⟨*short time pattern*⟩}

Again, when used with \TeXOSQuery, the short commands, such as \patdtf, are internally converted. They're not defined outside that scope.

⟨*current date time block*⟩

The current date and time provided in various formats suitable to the given locale:

{⟨*full date time*⟩}{⟨*long date time*⟩}{⟨*medium date time*⟩}{⟨*short date time*⟩}

This may simply be the date and time from above separated by a space.

⟨*date time pattern block*⟩

The pattern used to format the full date time, long date time, medium date time and short date time. This may simply be the date and time patterns from above separated by a space.

{⟨*full date time pattern*⟩}{⟨*long date time pattern*⟩}{⟨*medium date time pattern*⟩}{⟨*short date time pattern*⟩}

⟨*days of the week block*⟩

The week day names (starting with Monday for consistency with pgfcalendar) in the locale's language. Non-ASCII characters are marked up with \twrp.

{⟨*Monday*⟩}{⟨*Tuesday*⟩}{⟨*Wednesday*⟩}{⟨*Thursday*⟩}{⟨*Friday*⟩}{⟨*Saturday*⟩}{⟨*Sunday*⟩}

⟨*abbreviated days of the week block*⟩

   As above, but abbreviated.

⟨*month names block*⟩

   The month names.

⟨*abbreviated month names block*⟩

   The abbreviated month names.

⟨*standalone days of the week block*⟩

   The week day names when used in a standalone context (for example, a column
   header). This may be the same as the earlier ⟨*days of the week block*⟩ (and will
   be the same for `texosquery.jar` and `texosquery-jre5.jar`). The standalone
   support was introduced to Java 8.

⟨*abbreviated standalone days of the week block*⟩

   As above, but abbreviated.

⟨*standalone month names block*⟩

   The month names when used in a standalone context (for example, a column header).
   This may be the same as the earlier ⟨*month names block*⟩ (and will be the same
   for `texosquery.jar` and `texosquery-jre5.jar`). The standalone support was
   introduced to Java 8.

⟨*abbreviated standalone month names block*⟩

   As above but abbreviated.

⟨*numeric block*⟩

   The numeric data similar to `--numeric` but it's missing the ⟨*tag*⟩ (which is provided
   in the earlier ⟨*locale block*⟩) and there are two extra items:

   {⟨*group sep*⟩}{⟨*decimal sep*⟩}{⟨*exp sep*⟩}{⟨*use group*⟩}{⟨*currency
   code*⟩}{⟨*regional currency code*⟩}{⟨*currency sym*⟩}{⟨*currency tex*⟩}{⟨*currency
   sep*⟩} {⟨*percent sym*⟩}{⟨*per mill sym*⟩}

   See above for the elements that are also provided in `--numeric`. The additional
   elements are ⟨*percent sym*⟩ and ⟨*per mill sym*⟩, which are the percent and per-mill
   symbols, respectively. The percent symbol `%` has its category code changed to 12
   by `\TeXOSQuery`. As with other non-ASCII characters, the per-mill symbol will be
   marked up with `\twrp`.

⟨*numeric patterns block*⟩

   The patterns used to format decimals, integers, currency and percentages.

{⟨*decimal pattern*⟩}{⟨*integer pattern*⟩}{⟨*currency pattern*⟩}{⟨*percentages pattern*⟩}

As with the date and time patterns, when used with `\TeXOSQuery`, the short commands, such as `\patdgt`, are internally converted. They're not defined outside that scope.

If a pattern is used directly in the text, it will expand to the original pattern padded to ten digits. (Eleven digit integers are outside TeX's maximum number range.)

Any of these numeric patterns may be used in the first argument of the low-level user command `\texosqueryfmtnumber` described in section 3.1.5. This command uses the following macros:

`\texosquerypatfmtcurrencysign`

The currency sign (defaults to `\$`). For example, when parsing the previous ⟨*numeric block*⟩, this command can be redefined to the {⟨*currency sym*⟩} or {⟨*currency tex*⟩} elements.

`\texosquerypatfmtgroupsep`

The group separator (defaults to `,`). For example, when parsing the previous ⟨*numeric block*⟩, this command can be redefined to the {⟨*group sep*⟩} element.

`\texosquerypatfmtdecsep`

The decimal separator (defaults to `.`). For example, when parsing the previous ⟨*numeric block*⟩, this command can be redefined to the {⟨*dec sep*⟩} element.

`\texosquerypatfmtcurdecsep`

The monetary decimal separator (defaults to `.`). For example, when parsing the previous ⟨*numeric block*⟩, this command can be redefined to the {⟨*currency sep*⟩} element.

`\texosquerypatfmtexp`

The exponent sign (defaults to E). For example, when parsing the previous ⟨*numeric block*⟩, this command can be redefined to the {⟨*exp sep*⟩} element.

`\texosquerypatfmtpercentsign`

The percent symbol. For example, when parsing the previous ⟨*numeric block*⟩, this command can be redefined to the {⟨*percent sym*⟩} element.

```
\texosquerypatfmtpermillsign
```

The per-mill symbol. For example, when parsing the previous ⟨*numeric block*⟩, this command can be redefined to the {⟨*per-mill sym*⟩} element.

```
\texosquerypatfmticurrencysign
```

The international currency sign. This is defined as ¤ by default, unless the command \textcurrency has been defined, in which case that's used instead. If the UTF-8 character ¤ is available and \textcurrency hasn't been defined before texosquery was input, then you will need to redefine this command as appropriate.

```
\texosquerypatfmtminus
```

The minus sign.

```
\texosquerypatfmtplus
```

The plus sign.

2 texosquery.tex: generic TEX code

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

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

However, texosquery uses markup commands in some of the results which need to be defined first. The file texosquery.tex provides generic TEX code to do this for you and stores 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.

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

```
\TeXOSQuery{⟨cs⟩}{⟨args⟩}
```

where $\langle cs \rangle$ is the control sequence in which to store the result and $\langle args \rangle$ are the command line arguments to pass to texosquery. This first locally changes the category code of some problematic characters and defines the short markup commands that texosquery uses to identify characters that need to be interpreted literally (for example, in file names). These commands will automatically be expanded by \TeXOSQuery when the result is input. For example

```
texosquery -n
```

produces

```
\pdfd \fcln 20161129221559\fpls 00\fapo 00\fapo
```

but when used with

```
\TeXOSQuery{\result}{texosquery -n}
```

the \result command will be set to

```
D:20161129221559+00'00'
```

where the characters D : + and ' all have category code 12 (other).

If you're not able to use TeX's piped shell escape but you can run texosquery outside of TeX, then another approach is to first run texosquery with the output redirected to a temporary file and then build your document, but instead of using \TeXOSQuery, use:

\TeXOSQueryFromFile{$\langle cs \rangle$}{$\langle file\ name \rangle$}

where $\langle file\ name \rangle$ is the name of the temporary file. For example, first run

```
texosquery -b -n > tmpresult.tex
```

and then in the document:

```
\TeXOSQueryFromFile{\result}{tmpresult}
```

Avoid spaces and other awkward characters in the file name.

In the case of \TeXOSQuery, if the command failed, $\langle cs \rangle$ will be set to empty. This will also occur with \TeXOSQueryFromFile if the file doesn't exist. It's best to always test for success after use. For example:

```
\TeXOSQuery{\result}{-b}
\ifx\result\empty
 Failed!
\else
 Result: \result.
\fi
```

In the case of \TeXOSQuery, failure can occur because the dry run mode was on, or it can occur if the query was denied (for example, forbidden file access), or if there's a syntax error in the system call. In the case of \TeXOSQueryFromFile, the control sequence will be empty if the file doesn't exist or if the file was empty.

As from version 1.2, `texosquery` checks the `openin_any` setting, which may forbid read access. Java's security manager or the filing system may also forbid read access.

To determine the cause of the error, first inspect the log file to check if the shell escape was used. In the above example, if the shell escape was permitted, then the log file should include

```
(|texosquery -b)
```

Copy and paste the system call (`texosquery -b` in the above case) into your command prompt or terminal and insert the `--debug` switch at the start of the argument list. For example:

```
texosquery --debug -b
```

This should help determine whether it's a syntax error or a query forbidden by the operating system.

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.)

> **Important Note:** Take care of characters that have a special meaning to your shell. For example, bash interprets # as a comment. For example, if you have a file called `image#1.png`, then you can't simply do
>
> ```
> \TeXOSQuery{\result}{-p image#1.png}
> ```
>
> since bash will pass this as
>
> ```
> texosquery -p image
> ```
>
> (The `#1.png` part is treated as a comment.) Nor can you do
>
> ```
> \TeXOSQuery{\result}{-p image\#1.png}
> ```
>
> as TeX will replace the `\#` with # when passing the command invocation to the shell. The only way to deal with this situation is to do
>
> ```
> \TeXOSQuery{\result}{-p image\string\#1.png}
> ```

to protect the # character from both TeX and the shell.

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).

> **Important Note:** Remember that a query can still fail even if the dry run mode is off.

Note that if you switch off the dry run mode when the shell escape setting forbids the execution of texosquery, then you'll get the rather annoying error:

```
runpopen command not allowed: texosquery

! I can't find file `"|texosquery -b"'.
\TeXOSQueryInvoker ...TeXOSInvokerName \space #1"

\TeXOSQuery ...noexpand #1{\TeXOSQueryInvoker {#2}
                                              }}\@texosquery@tmp \fi
l.5 \TeXOSQuery{\result}{-b}
                           ^^M
(Press Enter to retry, or Control-D to exit)
```

By default, the dry run mode is only switched off if the unrestricted shell escape mode is on (detected through `\shellescape` or `\pdfshellescape`).

If texosquery is added to the restricted list, you can add

`\TeXOSQueryAllowRestricted`

to the `texosquery.cfg` file. (This command can't be used outside of that file.)

If you get the above error, then:

- make sure you don't have `\TeXOSQueryAllowRestricted` in your `texosquery.cfg` file;

- make sure you run TeX with the shell escape enabled;

- check the definition of `\TeXOSInvokerName`;

- try using the application directly from the command prompt or terminal. For example, in the above message, the bit between `` `"| `` and `"'` (that is, `texosquery-jre8 -b`) shows the attempted system call. Copy and paste it directly into your operating system's command prompt or terminal and to check the application has been installed correctly.

The `\TeXOSQuery` command uses `\TeXOSInvokerName` to reference the application name. This now defaults to `texosquery-jre8` (as from `texosquery` version 1.7) but needs to be redefined to reflect the particular system call that's required. For example, `texosquery` (Java 7) or `texosquery-jre5` (Java 5). This redefinition can be done in the configuration file `texosquery.cfg` for a system-wide setting. See section 1.1 for further details.

> **Important Note:** Some of the shortcut commands listed below require extra arguments after the relevant switch. These are automatically enclosed in single-quotes to protect any spaces, but only with the *unrestricted* shell escape. If the argument actually contains any single-quote characters, make sure you use `\string\'` to prevent interference. The quote character is forbidden in the shell escape in restricted mode (see section 1.5), so the arguments won't be delimited in this case and the quote character should be avoided.

> Since a file name reference 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 this shouldn't be an issue if you use a safe file naming scheme) and again be aware of the limitations imposed by the restricted shell escape.

> All paths should use a forward slash for the directory divider.

2.1 Locale

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}`

If you just want the codeset in the same form as `--locale-lcs` you can do:

`\TeXOSQuery{`⟨*cs*⟩`}{-C}`

Similarly for the unprocessed codeset (`--encoding`).

The IETF BCP 47 language tag (`-b` or `--bcp47`) can be obtained using:

`\TeXOSQueryLangTag{`⟨*cs*⟩`}`

The numeric separators and currency symbols (`-N` or `--numeric`) can be obtained using

`\TeXOSQueryNumeric{`⟨*cs*⟩`}{`⟨*locale*⟩`}`

The ⟨*locale*⟩ should be a valid language tag or may be empty for the system's default locale. Similarly for the command below.

All the locale data (`-D` or `--locale-data`) can be obtained using

`\TeXOSQueryLocaleData{`⟨*cs*⟩`}{`⟨*locale*⟩`}`

2.2 Operating System Information

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

`\TeXOSQueryName{⟨cs⟩}`

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

`\TeXOSQueryVersion{⟨cs⟩}`

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

`\TeXOSQueryArch{⟨cs⟩}`

2.3 Dates and Times

The current date and time information (`-M` or `--date-time`) can be obtained using:

`\TeXOSQueryDateTime{⟨cs⟩}`

Example usage:

`\texosquerydefpattern{\pattern}{\%2d/\%2M/\%4y \%2H:\%2m:\%2s}`

`\TeXOSQueryDateTime{\datetimedata}`

```
\ifx\datetimedata\empty
 Query Failed!
\else
 \expandafter\texosqueryfmtdatetime\expandafter\pattern\datetimedata
\fi
```

Note that commands such as `\texosqueryfmtpatMMM` will need to be defined to produce textual elements. See sections 3.1.3 and 3.1.4 for further details.

The time zone mappings (`-Z` or `--time-zones`) can be obtained using:

`\TeXOSQueryTimeZones{⟨cs⟩}{⟨locale⟩}`

Leave ⟨*locale*⟩ empty if the default locale is required.

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, XƎTeX).

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.

2.4 File Operations

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

`\TeXOSQueryCwd{`⟨*cs*⟩`}`

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

`\TeXOSQueryHome{`⟨*cs*⟩`}`

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

`\TeXOSQueryTmpDir{`⟨*cs*⟩`}`

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.

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.)

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

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

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

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.

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`.

To omit directories (`-ir` or `--list-regular`):

`\TeXOSQueryRegularFileList{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*dir*⟩`}`

To omit regular files (`-id` or `--list-dir`):

`\TeXOSQuerySubDirList{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*dir*⟩`}`

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}`

> **Important Note:** Unlike most of the return values the ⟨*sep*⟩ part here isn't escaped, so take care if ⟨*sep*⟩ contains any commands. For example, if you want to use `\\` as the separator, you'll need to use `\string\noexpand\string\\` in the ⟨*sep*⟩ part.
>
> `\TeXOSQueryFilterFileList{\result}{\string\noexpand\string\\}{.*\string\.tex}{.}`

If you want the list sorted, you can use the following which set the optional ⟨*sort*⟩ argument.

Order by last modified date starting with the oldest (`date-ascending`):

`\TeXOSQueryFileListDateAsc{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the regular files only list:

`\TeXOSQueryRegularFileListDateAsc{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the sub-directories only list:

`\TeXOSQuerySubDirListDateAsc{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or for the filtered list:

`\TeXOSQueryFilterFileListDateAsc{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

or for the filtered regular files only list:

`\TeXOSQueryFilterRegularFileListDateAsc{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

or for the filtered sub-directories only list:

`\TeXOSQueryFilterSubDirListDateAsc{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

Order by last modified date starting with the newest (`date-descending`):

`\TeXOSQueryFileListDateDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the regular files only list:

`\TeXOSQueryRegularFileListDateDes{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*dir*⟩`}`

or the sub-directories only list:

`\TeXOSQuerySubDirListDateDes{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*dir*⟩`}`

or for the filtered list:

`\TeXOSQueryFilterFileListDateDes{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*regex*⟩`}{`⟨*dir*⟩`}`

or the filtered regular files only list:

`\TeXOSQueryFilterRegularFileListDateDes{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*regex*⟩`}{`⟨*dir*⟩`}`

or the filtered sub-directories only list:

`\TeXOSQueryFilterRegularFileListDateDes{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*regex*⟩`}{`⟨*dir*⟩`}`

Order by file size starting with the smallest (`size-ascending`):

`\TeXOSQueryFileListSizeAsc{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*dir*⟩`}`

or the regular files only list:

`\TeXOSQueryRegularFileListSizeAsc{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*dir*⟩`}`

or the sub-directories only list:

`\TeXOSQuerySubDirListSizeAsc{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*dir*⟩`}`

or for the filtered list:

`\TeXOSQueryFilterFileListSizeAsc{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*regex*⟩`}{`⟨*dir*⟩`}`

or the filtered regular files only list:

`\TeXOSQueryFilterRegularFileListSizeAsc{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*regex*⟩`}{`⟨*dir*⟩`}`

or the filtered sub-directories only list:

`\TeXOSQueryFilterSubDirListSizeAsc{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*regex*⟩`}{`⟨*dir*⟩`}`

Order by file size starting with the largest (`size-descending`):

`\TeXOSQueryFileListSizeDes{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*dir*⟩`}`

or the regular files only list:

`\TeXOSQueryRegularFileListSizeDes{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*dir*⟩`}`

or the sub-directories only list:

`\TeXOSQuerySubDirListSizeDes{`⟨*cs*⟩`}{`⟨*sep*⟩`}{`⟨*dir*⟩`}`

or for the filtered list:

`\TeXOSQueryFilterFileListSizeDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the filtered regular files only list:

`\TeXOSQueryFilterRegularFileListSizeDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the filtered sub-directories only list:

`\TeXOSQueryFilterSubDirListSizeDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

Order by file name in alphabetical order (`name-ascending`):

`\TeXOSQueryFileListNameAsc{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the regular files only list:

`\TeXOSQueryRegularFileListNameAsc{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the sub-directories only list:

`\TeXOSQuerySubDirListNameAsc{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or for the filtered list:

`\TeXOSQueryFilterFileListNameAsc{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

or the filtered regular files only list:

`\TeXOSQueryFilterRegularFileListNameAsc{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

or the filtered sub-directories only list:

`\TeXOSQueryFilterSubDirListNameAsc{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

Order by file name in reverse alphabetical order (`name-descending`):

`\TeXOSQueryFileListNameDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the regular files only list:

`\TeXOSQueryRegularFileListNameDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the sub-directories only list:

`\TeXOSQuerySubDirListNameDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or for the filtered list:

`\TeXOSQueryFilterFileListNameDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the filtered regular files only list:

`\TeXOSQueryFilterRegularFileListNameDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the filtered sub-directories only list:

`\TeXOSQueryFilterSubDirListNameDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

Order by file name in case-insensitive alphabetical order (`iname-ascending`):

`\TeXOSQueryFileListNameIgnoreCaseAsc{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the regular files only list:

`\TeXOSQueryRegularFileListNameIgnoreCaseAsc{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the sub-directories only list:

`\TeXOSQuerySubDirListNameIgnoreCaseAsc{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or for the filtered list:

`\TeXOSQueryFilterFileListNameIgnoreCaseAsc{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

or the filtered regular files only list:

`\TeXOSQueryFilterRegularFileListNameIgnoreCaseAsc{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

or the filtered sub-directories only list:

`\TeXOSQueryFilterSubDirListNameIgnoreCaseAsc{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

Order by file name in reverse case-insensitive alphabetical order (`iname-descending`):

`\TeXOSQueryFileListNameIgnoreCaseDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the regular files only list:

`\TeXOSQueryRegularFileListNameIgnoreCaseDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the sub-directories only list:

`\TeXOSQuerySubDirListNameIgnoreCaseDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or for the filtered list:

`\TeXOSQueryFilterFileListNameIgnoreCaseDes{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

or the filtered regular files only list:

`\TeXOSQueryFilterRegularFileListNameIgnoreCaseDes{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

or the filtered sub-directories only list:

`\TeXOSQueryFilterSubDirListNameIgnoreCaseDes{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

Order by file extension in alphabetical order (`ext-ascending`):

`\TeXOSQueryFileListExtAsc{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the regular files only list:

`\TeXOSQueryRegularFileListExtAsc{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the sub-directories only list:

`\TeXOSQuerySubDirListExtAsc{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or for the filtered list:

`\TeXOSQueryFilterFileListExtAsc{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

or the filtered regular files only list:

`\TeXOSQueryFilterRegularFileListExtAsc{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

or the filtered sub-directories only list:

`\TeXOSQueryFilterSubDirListExtAsc{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

Order by file extension in reverse alphabetical order (`ext-descending`):

`\TeXOSQueryFileListExtDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the regular files only list:

`\TeXOSQueryRegularFileListExtDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or the sub-directories only list:

`\TeXOSQuerySubDirListExtDes{⟨cs⟩}{⟨sep⟩}{⟨dir⟩}`

or for the filtered list:

`\TeXOSQueryFilterFileListExtDes{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

or the filtered regular files only list:

`\TeXOSQueryFilterRegularFileListExtDes{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

or the filtered sub-directories only list:

`\TeXOSQueryFilterSubDirListExtDes{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

A recursive filtered list of regular files starting from a given directory on the current working path (`-w` or `--alk`) can be obtained using:

`\TeXOSQueryWalk{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

where ⟨*regex*⟩ is as for the filtered listings described above.

To sort according to last modified date:

`\TeXOSQueryWalkDateAsc{⟨cs⟩}{⟨sep⟩}{⟨regex⟩}{⟨dir⟩}`

or in reverse order:

\TeXOSQueryWalkDateDes{⟨*cs*⟩}{⟨*sep*⟩}{⟨*regex*⟩}{⟨*dir*⟩}

To sort according to file size:

\TeXOSQueryWalkSizeAsc{⟨*cs*⟩}{⟨*sep*⟩}{⟨*regex*⟩}{⟨*dir*⟩}

or in reverse order:

\TeXOSQueryWalkSizeDes{⟨*cs*⟩}{⟨*sep*⟩}{⟨*regex*⟩}{⟨*dir*⟩}

To sort according to path name (case-sensitive):

\TeXOSQueryWalkNameAsc{⟨*cs*⟩}{⟨*sep*⟩}{⟨*regex*⟩}{⟨*dir*⟩}

or in reverse order:

\TeXOSQueryWalkNameDes{⟨*cs*⟩}{⟨*sep*⟩}{⟨*regex*⟩}{⟨*dir*⟩}

To sort according to path name (case-insensitive):

\TeXOSQueryWalkNameIgnoreCaseAsc{⟨*cs*⟩}{⟨*sep*⟩}{⟨*regex*⟩}{⟨*dir*⟩}

or in reverse order:

\TeXOSQueryWalkNameIgnoreCaseDes{⟨*cs*⟩}{⟨*sep*⟩}{⟨*regex*⟩}{⟨*dir*⟩}

To sort according to file extension:

\TeXOSQueryWalkExtAsc{⟨*cs*⟩}{⟨*sep*⟩}{⟨*regex*⟩}{⟨*dir*⟩}

or in reverse order:

\TeXOSQueryWalkExtDes{⟨*cs*⟩}{⟨*sep*⟩}{⟨*regex*⟩}{⟨*dir*⟩}

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 \else
10   \@texosquery@restore@at
11   \expandafter\endinput
12 \fi
```

Version info.

13 `\expandafter\def\csname ver@texosquery.tex\endcsname{2020/02/04 v1.7 (NLCT)}`

`\@texosquery@warn` Generate warning message. Use tracklang's warning if available (so that the warnings can be disabled for both packages at the same time).

```
14 \ifx\@tracklang@pkgwarn\undefined
15   \ifx\PackageWarning\undefined
16     \def\@texosquery@warn#1{%
17       {%
18         \newlinechar='\^^J
19         \def\MessageBreak{^^J}%
20         \message{^^Jtexosquery Warning: #1 on line \the\inputlineno.^^J}%
21       }%
22     }
23   \else
24     \def\@texosquery@warn#1{%
25       \PackageWarning{texosquery}{#1}%
26     }
27   \fi
28 \else
29   \def\@texosquery@warn#1{%
30     \@tracklang@pkgwarn{texosquery}{#1}%
31   }
32 \fi
```

`\@texosquery@info`

```
33 \ifx\PackageInfo\undefined
34   \def\@texosquery@info#1{%
35     {%
36       \newlinechar='\^^J
37       \def\MessageBreak{^^J}%
38       \message{^^Jtexosquery Info: #1^^J}%
39     }%
40   }
41 \else
42   \def\@texosquery@info#1{%
43     \PackageInfo{texosquery}{#1}%
44   }
45 \fi
```

`\@texosquery@err`

```
46 \ifx\PackageError\undefined
47   \def\@texosquery@err#1#2{%
48     \errhelp{#2}%
49     \errmessage{texosquery: #1}}
50 \else
51   \def\@texosquery@err#1#2{\PackageError{texosquery}{#1}{#2}}
52 \fi
```

`\@texosquery@ifundef`   This is defined in the same way as tracklang's `\@tracklang@ifundef`. (Can't assume tracklang has been loaded.)

```
53 \long\def\@texosquery@ifundef#1#2#3{%
54   \ifcsname#1\endcsname
55     \expandafter\ifx\csname #1\endcsname\relax
56       #2%
57     \else
58       #3%
59     \fi
60   \else
61     \expandafter\ifx\csname #1\endcsname\relax
62       #2%
63     \else
64       #3%
65     \fi
66   \fi
67 }
68 \ifx\ifcsname\undefined
69   \long\def\@texosquery@ifundef#1#2#3{%
70     \expandafter\ifx\csname #1\endcsname\relax
71       #2%
72     \else
73       #3%
74     \fi
75   }
76 \fi
```

`\TeXOSInvokerName`   The name of the texosquery application.

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

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

```
78 \ifx\@@input\undefined
79   \let\texosquery@input\input
80 \else
81   \let\texosquery@input\@@input
82 \fi
```

`\TeXOSQueryInvoker`   Invoke the application in the piped shell escape with the given arguments.

```
83 \def\TeXOSQueryInvoker#1{\texosquery@input|"\TeXOSInvokerName\space#1" }
```

`\ifTeXOSQueryDryRun`   Provide a dry-run mode.

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

`\TeXOSQueryAllowRestricted`   The default behaviour only switches off the dry-run mode if the shell escape is unrestricted. The configuration file may override this with `\TeXOSQueryAllowRestricted`, which will allow the dry run mode to be switched off if restricted mode is detected.

```
85 \def\TeXOSQueryAllowRestricted{%
86   \def\@texosquery@allowrestricted##1##2{##1}%
87 }
```

\TeXOSQueryDenyRestricted Switch it off.

```
88 \def\TeXOSQueryDenyRestricted{%
89  \def\@texosquery@allowrestricted##1##2{##2}%
90 }
```

\@texosquery@allowrestricted Initialise to prevent shell escape in restricted mode.

```
91 \def\@texosquery@allowrestricted#1#2{#2}%
```

\TeXOSInvokerRestrictedMessage Message used if TeX is run in restricted mode but the invoker name hasn't been allowed. This may be redefined in the configuration file to customise the message if the invoker name isn't allowed. Set to empty or \relax if the message should be omitted.

```
92 \def\TeXOSInvokerRestrictedMessage{%
93  \string\TeXOSQuery\space doesn't work in dry run mode.
94  \MessageBreak
95  If '\TeXOSInvokerName' has been added to the \MessageBreak
96  restricted list, make sure that
97  \MessageBreak \string\TeXOSQueryAllowRestricted\space
98  \MessageBreak has been uncommented in the configuration file
99  \MessageBreak (texosquery.cfg) otherwise you need to use the
100 \MessageBreak unrestricted shell escape}
```

Only need to write this information to the transcript if \TeXOSQuery is actually used and the dry run mode is on in restricted mode.

@texosquery@dorestrictedmessage

```
101 \let\@texosquery@dorestrictedmessage\relax
```

Load the configuration file if it exists.

```
102 \openin0=texosquery.cfg \ifeof0\relax \else
103   \closein0\relax
104   \begingroup
105     \@texosquery@info{reading configuration file}%
106   \endgroup
107   \input texosquery.cfg
108 \fi
```

Disable cfg-only commands:

```
109 \def\TeXOSQueryAllowRestricted{%
110   \@texosquery@warn{\string\TeXOSQueryAllowRestricted\space
111     ignored (only allowed in texosquery.cfg)}%
112 }
113 \def\TeXOSQueryDenyRestricted{%
114   \@texosquery@warn{\string\TeXOSQueryDenyRestricted\space
115     ignored (only allowed in texosquery.cfg)}%
116 }
```

\texosquery@shellescape We need to know the shell escape mode. This is provided by PDFTeX's \pdfshellescape primitive or XeTeX's \shellescape primitive. It's a little more complicated in LuaTeX. Initialise to 0:

```
117 \chardef\texosquery@shellescape=0
```

45

Test for the PDFTEX and XƎTEX primitives.

```
118 \ifx\shellescape\undefined
119   \ifx\pdfshellescape\undefined
```

Neither primitive defined. Has pdftexcmds been loaded? If so, we can use `\pdf@shellescape`.

```
120     \ifx\pdf@shellescape\undefined
```

`\pdf@shellescape` hasn't been defined. Is LuaTEX in use?

```
121       \ifx\directlua\undefined
```

Not LuaTEX. Most likely an old TEX format. Have to assume the shell escape is unavailable.

```
122       \else
```

Use Lua to find the shell escape. This uses the same method as pdftexcmds. (Not loading that package, as this is the only thing required from it.)

```
123         \edef\texosquery@shellescape{\directlua0{
124           if os.execute then
125             if status
126                and status.luatex_version
127                and status.luatex_version >= 68 then
128               tex.write(os.execute())
129             else
130               local result = os.execute()
131               if result == 0 then
132                 tex.write("0")
133               else
134                 if result == nil then
135                   tex.write("0")
136                 else
137                   tex.write("1")
138                 end
139               end
140             end
141           else
142             tex.write("0")
143           end
144         }}
145       \fi
146     \else
147       \let\texosquery@shellescape\pdf@shellescape
148     \fi
149   \else
```

PDFTEX's `\pdfshellescape` has been defined, so that can be used, but first check it hasn't been set to `\relax`.

```
150     \ifx\pdfshellescape\relax
```

Something's meddled with `\pdfshellescape`, so try `\pdf@shellescape`.

```
151       \ifx\pdf@shellescape\undefined
```

`\pdf@shellescape` hasn't been defined. Is LuaTeX in use?

```
152          \ifx\directlua\undefined
```

Not LuaTeX. Most likely an old TeX format. Have to assume the shell escape is unavailable.

```
153          \else
```

Use Lua to find the shell escape.

```
154              \edef\texosquery@shellescape{\directlua0{
155            if os.execute then
156              if status
157                  and status.luatex_version
158                  and status.luatex_version >= 68 then
159                tex.write(os.execute())
160              else
161                local result = os.execute()
162                if result == 0 then
163                  tex.write("0")
164                else
165                  if result == nil then
166                    tex.write("0")
167                  else
168                    tex.write("1")
169                  end
170                end
171              end
172            else
173              tex.write("0")
174            end
175          }}
176        \fi
177      \else
178        \let\texosquery@shellescape\pdf@shellescape
179      \fi
180    \else
181      \let\texosquery@shellescape\pdfshellescape
182    \fi
183  \fi
184 \else
```

X̲ETEX's `\shellescape` has been defined, so that can be used, but first check it hasn't been set to `\relax`.

```
185    \ifx\shellescape\relax
```

Something's meddled with `\shellescape`, so try `\pdfshellescape`.

```
186      \ifx\pdfshellescape\undefined
187      \else
```

`\pdfshellescape` has been defined, so that can be used, but first check it hasn't been set to `\relax`.

```
188        \ifx\pdfshellescape\relax
```

Something's meddled with \pdfshellescape, so try \pdf@shellescape.

```
189         \ifx\pdf@shellescape\undefined
```

Try Lua.

```
190            \ifx\directlua\undefined
```

Not LuaTEX. Most likely an old TEX format. Have to assume the shell escape is unavailable.

```
191            \else
```

Use Lua to find the shell escape.

```
192               \edef\texosquery@shellescape{\directlua0{
193                 if os.execute then
194                   if status
195                      and status.luatex_version
196                      and status.luatex_version >= 68 then
197                    tex.write(os.execute())
198                  else
199                    local result = os.execute()
200                    if result == 0 then
201                      tex.write("0")
202                    else
203                      if result == nil then
204                        tex.write("0")
205                      else
206                        tex.write("1")
207                      end
208                    end
209                  end
210                else
211                  tex.write("0")
212                end
213               }}
214          \fi
215        \else
216          \let\texosquery@shellescape\pdf@shellescape
217        \fi
218      \else
219        \let\texosquery@shellescape\pdfshellescape
220      \fi
221    \fi
222  \else
223    \let\texosquery@shellescape\shellescape
224  \fi
225 \fi
```

If shell escape is unrestricted, automatically switch off dry-run mode, unless the cfg file has allowed it.

```
226 \ifcase\texosquery@shellescape
227   \@texosquery@info{shell escape disabled, dry-run mode on}
```

```
228 \or
229   \TeXOSQueryDryRunfalse
230 \or
231   \@texosquery@allowrestricted
232   {%
233     \TeXOSQueryDryRunfalse
234   }
235   {%
236     \@texosquery@info{shell escape restricted, dry-run mode on.}
```

Enable the restricted warning message, but it only needs to be written once, so disable it after use.

```
237     \def\@texosquery@dorestrictedmessage{%
238       \ifx\TeXOSInvokerRestrictedMessage\empty
239       \else
240         \ifx\TeXOSInvokerRestrictedMessage\relax
241         \else
242           \@texosquery@warn{\TeXOSInvokerRestrictedMessage}%
243         \fi
244       \fi
245       \let\@texosquery@dorestrictedmessage\relax
246     }
247   }
248 \fi
```

`\@texosquery@edef`   Need to provide some protection (if available) against non-ASCII characters that have been made active by inputenc when reading in the results of the shell escape. This command may be defined before loading texosquery, otherwise it's set to \protected@edef, if defined, or \edef.

```
249 \ifx\@texosquery@edef\undefined
250   \ifx\protected@edef\undefined
251     \let\@texosquery@edef\edef
252   \else
253     \let\@texosquery@edef\protected@edef
254   \fi
255 \fi
```

Provide some utility commands. (Can't use \@gobble etc, as we may not be using LaTeX.)

`\@texosquery@gobble`

```
256 \def\@texosquery@gobble#1{}
```

`\@texosquery@firstofone`

```
257 \def\@texosquery@firstofone#1{#1}
```

The results obtained from texosquery may be file names for use in commands like \input or \includegraphics or they may be text that needs typesetting (such as month

names) or they may be date-time patterns or numeric patterns or they may be PDF date-time strings, which may need to have the category code of the initial "D" set to 12 for parsing commands that include this character in the argument syntax.

This means that we need to take special characters into account, but the way they are dealt with vary according to context. For example, # needs to have the category code set to 12 if it's part of a file name. If an image file is called, say, `test_imagefile#.png` then the following doesn't work:

`\includegraphics{test\_imagefile\#}`

It needs to be

`\includegraphics{test\string_imagefile\string#}`

or

`\includegraphics{\detokenize{test_imagefile#}}`

The first two versions of texosquery try to deal with this by simply changing the category code of _ to 12 and getting `texosquery` to replace all instances of # with \#. This hash substitution doesn't work with the above image example so version 1.2 introduced a new command that `texosquery` could use instead of \# that expands to `\string#`. This now solves the problem for file names that are obtained through `texosquery`, but `texosquery` doesn't solely return file names. It also returns text that needs typesetting and it also returns numeric patterns, which in their raw form include # as a digit identifier.

This means that we can't simply detokenize the result from `texosquery`. Instead `texosquery` replaces problematic characters with control sequences *according to context*. For example, `\texosqueryhash` is used in a file name context, `\texosquerytexthash` is used in a textual context and `\texosquerypatdigitnozero` in a numeric pattern context.

These long control sequence names clutter the results when testing the application directly in a terminal, so the Java code uses short forms that are locally defined by `\TeXOSQuery` to expand to the longer forms.

`\texosquerynonasciiwrap` Allow a way to deal with non-ASCII characters returned by `texosquery`. `\TeXOSQuery` locally defines `\twrp` to this command. By default this just does its argument but may be redefined. For example, if the document uses a different file encoding to Java, then this command might need to be redefined to perform the appropriate conversion.

258 `\def\texosquerynonasciiwrap#1{#1}`

`\texosquerynonasciidetokwrap` We also need to allow for non-ASCII characters in file names. In this case the argument needs detokenizing. With eTeX, we can simply use `\detokenize` but we need to allow for plain non-extended TeX, so check for the existence of `\detokenize` first.

259 `\ifx\detokenize\undefined`

This won't work for characters consisting of multiple octets, but if users want UTF-8 support then they really need eTeX at the very least (but ideally XeTeX or LuaTeX).

260 `   \def\texosquerynonasciidetokwrap#1{\string#1}`
261 `\else`
262 `   \def\texosquerynonasciidetokwrap#1{\detokenize{#1}}`
263 `\fi`

50

Now define commands used in \TeXOSQuery for various escaped characters. The literal versions are for file names. The textual versions are for use within the document text. For completeness, all the ASCII punctuation characters have both a literal and textual version. This helps to protect against babel shorthands etc.

\texosquerybackslash   Literal backslash.

264 \edef\texosquerybackslash{\expandafter\@texosquery@gobble\string\\}

\texosquerytextbackslash   Textual backslash.

```
265 \ifx\textbackslash\undefined
266   \def\texosquerytextbackslash{\texosquerybackslash}
267 \else
268   \def\texosquerytextbackslash{\noexpand\textbackslash}
269 \fi
```

\texosqueryleftbrace   Literal left brace.

270 \edef\texosqueryleftbrace{\expandafter\@texosquery@gobble\string\{}

\texosquerytextleftbrace   Textual left brace.

271 \def\texosquerytextleftbrace{\{}

\texosqueryrightbrace   Literal right brace.

272 \edef\texosqueryrightbrace{\expandafter\@texosquery@gobble\string\}}

\texosquerytextrightbrace   Textual right brace.

273 \def\texosquerytextrightbrace{\}}

\texosqueryhash   Literal hash.

274 \edef\texosqueryhash{\expandafter\@texosquery@gobble\string\#}

\texosquerytexthash   Textual hash.

275 \def\texosquerytexthash{\#}

\texosqueryunderscore   Literal underscore.

276 \edef\texosqueryunderscore{\expandafter\@texosquery@gobble\string_}

\texosquerytextunderscore   Textual underscore.

277 \def\texosquerytextunderscore{_}

\texosquerybacktick   Literal grave.

278 \edef\texosquerybacktick{\string`}

\texosquerytextbacktick   Textual open quote.

279 \def\texosquerytextbacktick{`}

\texosqueryclosequote   Literal apostrophe.

280 \edef\texosqueryclosequote{\string'}

`\texosquerytextclosequote`	Textual apostrophe / single closing quote.
	281 `\def\texosquerytextclosequote{'}`
`\texosquerydoublequote`	Literal double-quote.
	282 `\edef\texosquerydoublequote{\string"}`
`\texosquerytextdoublequote`	Textual double-quote.
	283 `\def\texosquerytextdoublequote{"}`
`\texosquerycolon`	Literal colon.
	284 `\edef\texosquerycolon{\string:}`
`\texosquerytextcolon`	Textual colon.
	285 `\def\texosquerytextcolon{:}`
`\texosquerysemicolon`	Literal semi-colon.
	286 `\edef\texosquerysemicolon{\string;}`
`\texosquerytextsemicolon`	Textual semi-colon.
	287 `\def\texosquerytextsemicolon{;}`
`\texosqueryequals`	Literal equals.
	288 `\edef\texosqueryequals{\string=}`
`\texosquerytextequals`	Textual equals.
	289 `\def\texosquerytextequals{=}`
`\texosqueryslash`	Literal slash.
	290 `\edef\texosqueryslash{\string/}`
`\texosquerytextslash`	Textual slash.
	291 `\def\texosquerytextslash{/}`
`\texosqueryhyphen`	Literal hyphen.
	292 `\edef\texosqueryhyphen{\string-}`
`\texosquerytexthyphen`	Textual hyphen.
	293 `\def\texosquerytexthyphen{-}`
`\texosqueryplus`	Literal plus.
	294 `\edef\texosqueryplus{\string+}`
`\texosquerytextplus`	Textual plus.
	295 `\def\texosquerytextplus{+}`
`\texosqueryperiod`	Literal period.
	296 `\edef\texosqueryperiod{\string.}`

\texosquerytextperiod	Textual period.
	297 \def\texosquerytextperiod{.}
\texosquerycomma	Literal comma.
	298 \edef\texosquerycomma{\string,}
\texosquerytextcomma	Textual comma.
	299 \def\texosquerytextcomma{,}
\texosqueryopenparen	Literal open bracket.
	300 \edef\texosqueryopenparen{\string(}
\texosquerytextopenparen	Textual open bracket.
	301 \def\texosquerytextopenparen{(}
\texosquerycloseparen	Literal close bracket.
	302 \edef\texosquerycloseparen{\string)}
\texosquerytextcloseparen	Textual close bracket.
	303 \def\texosquerytextcloseparen{)}
\texosqueryopensq	Literal open square bracket.
	304 \edef\texosqueryopensq{\string[}
\texosquerytextopensq	Textual open square bracket.
	305 \def\texosquerytextopensq{[}
\texosqueryclosesq	Literal close square bracket.
	306 \edef\texosqueryclosesq{\string]}
\texosquerytextclosesq	Textual close square bracket.
	307 \def\texosquerytextclosesq{]}
\texosqueryasterisk	Literal asterisk.
	308 \edef\texosqueryasterisk{\string*}
\texosquerytextasterisk	Textual asterisk.
	309 \def\texosquerytextasterisk{*}
\texosqueryatchar	Literal at character.
	310 \edef\texosqueryatchar{\string @}
\texosquerytextatchar	Textual at character.
	311 \def\texosquerytextatchar{@}
\texosquerybar	Literal bar.
	312 \edef\texosquerybar{\string|}

\texosquerytextbar	Textual bar.

```
313 \ifx\undefined\textbar
314   \def\texosquerytextbar{|}
315 \else
316   \def\texosquerytextbar{\ifmmode|\else\textbar\fi}
317 \fi
```

\texosquerylessthan	Literal less than.

```
318 \edef\texosquerylessthan{\string<}
```

\texosquerytextlessthan	Textual less than.

```
319 \ifx\undefined\textless
320   \def\texosquerytextlessthan{<}
321 \else
322   \def\texosquerytextlessthan{\ifmmode<\else\textless\fi}
323 \fi
```

\texosquerygreaterthan	Literal greater than.

```
324 \edef\texosquerygreaterthan{\string>}
```

\texosquerytextgreaterthan	Textual greater than.

```
325 \ifx\undefined\textgreater
326   \def\texosquerytextgreaterthan{>}
327 \else
328   \def\texosquerytextgreaterthan{\ifmmode<\else\textgreater\fi}
329 \fi
```

\texosquerytilde	Literal tilde.

```
330 \edef\texosquerytilde{\string~}
```

\texosquerytexttilde	Textual tilde.

```
331 \ifx\textasciitilde\undefined
332   \def\texosquerytexttilde{\string~}
333 \else
334   \def\texosquerytexttilde{\textasciitilde}
335 \fi
```

\texosquerycircum	Literal circumflex.

```
336 \edef\texosquerycircum{\string^}
```

\texosquerytextcircum	Textual circumflex.

```
337 \ifx\textasciicircum\undefined
338   \def\texosquerytextcircum{\string^}
339 \else
340   \def\texosquerytextcircum{\textasciicircum}
341 \fi
```

\texosqueryampersand	Literal ampersand.

```
342 \edef\texosqueryampersand{\string&}
```

`\texosquerytextampersand`	Textual ampersand.

343 `\def\texosquerytextampersand{\&}`

`\texosquerydollar`	Literal dollar. (This could just be defined as `\string$`, but that plays havoc with the syntax highlighting!)

344 `\edef\texosquerydollar{\expandafter\@texosquery@gobble\string\$}`

`\texosquerytextdollar`	Textual dollar.

345 `\def\texosquerytextdollar{\$}`

`\texosquerypercent`	Literal percent.

346 `\edef\texosquerypercent{\expandafter\@texosquery@gobble\string\%}`

`\texosquerytextpercent`	Textual percent.

347 `\def\texosquerytextpercent{\%}`

`\texosqueryexclam`	Literal exclamation.

348 `\edef\texosqueryexclam{\string!}`

`\texosquerytextexclam`	Textual exclamation.

349 `\def\texosquerytextexclam{!}`

`\texosqueryquestion`	Literal question mark.

350 `\edef\texosqueryquestion{\string?}`

`\texosquerytextquestion`	Textual question mark.

351 `\def\texosquerytextquestion{?}`

`\texosqueryliteralspace`	Literal space.

352 `\edef\texosqueryliteralspace{\expandafter\string\space}`

`\texosquerytextspace`	Textual space. (Don't allow it to expand while it's being fetched from `\texosquery` just in case it disappears.)

353 `\def\texosquerytextspace{\noexpand\space}`

`\@texosquery@D`	The D identifier in PDF date-time formats need to have category code 12. This is only used by methods that return results in the form:

$$\texttt{D:}\langle YYYY \rangle \langle MM \rangle \{\langle DD \rangle\} \langle HH \rangle \langle mm \rangle \langle ss \rangle \langle TZh \rangle' \langle TZm \rangle'$$

354 `\edef\@texosquery@D{\string D}`

`\@texosquery@enableshortcs`	Enable shortcut commands.

355 `\def\@texosquery@enableshortcs{%`

Thesre are for the date-time and numeric patterns.

```
356    \def\patdtf{\noexpand\texosquerydtf}%
357    \def\patpmnumfmt{\noexpand\texosquerypatplusminus}%
358    \def\patnumfmt{\noexpand\texosquerypatnum}%
359    \def\patsinumfmt{\noexpand\texosquerypatsinum}%
360    \def\patdecfmt{\noexpand\texosquerypatdec}%
361    \def\patpcur{\noexpand\texosquerypatprefixcurrency}%
362    \def\patpicur{\noexpand\texosquerypatprefixicurrency}%
363    \def\patscur{\noexpand\texosquerypatsuffixcurrency}%
364    \def\patsicur{\noexpand\texosquerypatsuffixicurrency}%
365    \def\patstr{\noexpand\texosquerypatstr}%
366    \def\patapo{\noexpand\texosquerypatquote}%
367    \def\patdgt{\noexpand\texosquerypatdigit}%
368    \def\patdgtnz{\noexpand\texosquerypatdigitnozero}%
369    \def\patmsg{\noexpand\texosquerypatminus}%
370    \def\patngp{\noexpand\texosquerypatgroupsep}%
371    \def\patppct{\noexpand\texosquerypatprefixpercent}%
372    \def\patspct{\noexpand\texosquerypatsuffixpercent}%
373    \def\patppml{\noexpand\texosquerypatprefixpermill}%
374    \def\patspml{\noexpand\texosquerypatsuffixpermill}%
```

Hook to adjust the processing of non-ASCII characters.

```
375    \def\twrp{\texosquerynonasciiwrap}%
376    \def\fwrp{\texosquerynonasciidetokwrap}%
```

Locally redefine some more commands that may occur in `texosquery`'s return value (via the `escapeSpChars` method in `TeXOSQuery.java`). The `t` prefix indicates textual commands and the `f` prefix indicates literal characters, for example, in file names.

```
377    \let\fbks\texosquerybackslash
378    \let\tbks\texosquerytextbackslash
379    \let\flbr\texosqueryleftbrace
380    \let\tlbr\texosquerytextleftbrace
381    \let\frbr\texosqueryrightbrace
382    \let\trbr\texosquerytextrightbrace
383    \let\fhsh\texosqueryhash
384    \let\thsh\texosquerytexthash
385    \let\fusc\texosqueryunderscore
386    \let\tusc\texosquerytextunderscore
387    \let\fgrv\texosquerybacktick
388    \let\tgrv\texosquerytextbacktick
389    \let\fapo\texosqueryclosequote
390    \let\tapo\texosquerytextclosequote
391    \let\fdqt\texosquerydoublequote
392    \let\tdqt\texosquerytextdoublequote
393    \let\fspc\texosqueryliteralspace
394    \let\tspc\texosquerytextspace
395    \let\fcln\texosquerycolon
396    \let\tcln\texosquerytextcolon
397    \let\fscl\texosquerysemicolon
398    \let\tscl\texosquerytextsemicolon
```

```
399        \let\feql\texosqueryequals
400        \let\teql\texosquerytextequals
401        \let\fhyn\texosqueryhyphen
402        \let\thyn\texosquerytexthyphen
403        \let\fpls\texosqueryplus
404        \let\tpls\texosquerytextplus
405        \let\ftld\texosquerytilde
406        \let\ttld\texosquerytexttilde
407        \let\fcir\texosquerycircum
408        \let\tcir\texosquerytextcircum
409        \let\famp\texosqueryampersand
410        \let\tamp\texosquerytextampersand
411        \let\fslh\texosqueryslash
412        \let\tslh\texosquerytextslash
413        \let\fpct\texosquerypercent
414        \let\tpct\texosquerytextpercent
415        \let\fexc\texosqueryexclam
416        \let\texc\texosquerytextexclam
417        \let\fque\texosqueryquestion
418        \let\tque\texosquerytextquestion
419        \let\fles\texosquerylessthan
420        \let\tles\texosquerytextlessthan
421        \let\fgre\texosquerygreaterthan
422        \let\tgre\texosquerytextgreaterthan
423        \let\fdol\texosquerydollar
424        \let\tdol\texosquerytextdollar
425        \let\fdot\texosqueryperiod
426        \let\tdot\texosquerytextperiod
427        \let\fcom\texosquerycomma
428        \let\tcom\texosquerytextcomma
429        \let\fopb\texosqueryopenparen
430        \let\topb\texosquerytextopenparen
431        \let\fclb\texosquerycloseparen
432        \let\tclb\texosquerytextcloseparen
433        \let\fosb\texosqueryopensq
434        \let\tosb\texosquerytextopensq
435        \let\fcsb\texosqueryclosesq
436        \let\tcsb\texosquerytextclosesq
437        \let\fast\texosqueryasterisk
438        \let\tast\texosquerytextasterisk
439        \let\fatc\texosqueryatchar
440        \let\tatc\texosquerytextatchar
441        \let\pdfd\@texosquery@D
442 }
```

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

```
443 \def\TeXOSQuery#1#2{%
444    \ifTeXOSQueryDryRun
445        \@texosquery@dorestrictedmessage
```

```
446    \begingroup
447      \newlinechar='\^^J
448      \message{^^JTeXOSQuery: \TeXOSInvokerName\space#2^^J}%
449    \endgroup
450    \def#1{}%
451  \else
452    \begingroup
453    \endlinechar=-1\relax
```

Locally redefine short commands used by texosquery

```
454    \@texosquery@enableshortcs
```

Change the category code of some potentially awkward characters. (This should no longer be an issue with the new commands that are now used in the returned text, but texosquery might be run with the backward compatibility mode on, so this is still needed just in case.)

```
455    \catcode'\-=12\relax
456    \catcode'\_=12\relax
457    \catcode'\^=12\relax
458    \catcode'\~=12\relax
459    \catcode'\$=12\relax
460    \catcode'\&=12\relax
461    \catcode'\.=12\relax
462    \catcode'\/=12\relax
463    \catcode'\:=12\relax
464    \catcode'\"=12\relax
465    \catcode'\'=12\relax
466    \catcode'\;=12\relax
467    \catcode'\%=12\relax
468    \everyeof{\noexpand}\relax
469    \@texosquery@edef\@texosquery@tmp{\endgroup\def\noexpand#1{\TeXOSQueryInvoker{#2}}}\@texosqu
470  \fi
471 }
```

\TeXOSQueryFromFile This is like \TeXOSQuery but doesn't use the piped shell escape. Instead, it's for cases where texosquery is run before TeX and the results have been captured in a file. For example, texosquery -b > texosqueryresult.tex The second argument is the file name.

```
472 \def\TeXOSQueryFromFile#1#2{%
```

First check that the file exists.

```
473    \openin0=#2
474      \ifeof0\relax
475        \def#1{}%
476      \else
477        \closein0\relax
478        \begingroup
479        \endlinechar=-1\relax
480        \@texosquery@enableshortcs
481        \catcode'\-=12\relax
482        \catcode'\_=12\relax
```

```
483      \catcode'\^=12\relax
484      \catcode'\~=12\relax
485      \catcode'\$=12\relax
486      \catcode'\&=12\relax
487      \catcode'\.=12\relax
488      \catcode'\/=12\relax
489      \catcode'\:=12\relax
490      \catcode'\"=12\relax
491      \catcode'\'=12\relax
492      \catcode'\;=12\relax
493      \catcode'\%=12\relax
494      \everyeof{\noexpand}\relax
495      \@texosquery@edef\@texosquery@tmp{\endgroup\def\noexpand#1{\texosquery@input #2 }}\@texos
496   \fi
497 }
```

3.1.1 Currency

The ⟨*T_EX currency*⟩ element of `--numeric` and `--locale-data` identifies the currency
symbol using

\texosquerycurrency{⟨*label*⟩}

which simply expands to the appropriate command.

\texosquerycurrency

```
498 \def\texosquerycurrency#1{%
499 \expandafter\noexpand\csname texosquerycurrency#1\endcsname
500 }
```

Provide the currency commands that may be returned texosquery (on expansion of
\texosquerycurrency). Most of these will need redefining as there's no appropriate
generic code to use as a default. The fontawesome package has the most support for
currency symbols, so these are checked first.

\texosquerycurrencydollar

```
501 \ifx\faDollar\undefined
502   \def\texosquerycurrencydollar{\$}
503 \else
504   \def\texosquerycurrencydollar{\faDollar}
505 \fi
```

\texosquerycurrencycent

```
506 \ifx\textcent\undefined
507   \def\texosquerycurrencycent{cent}
508 \else
509   \def\texosquerycurrencycent{\textcent}
510 \fi
```

`\texosquerycurrencypound`

```
511 \ifx\faGbp\undefined
512   \ifx\pounds\undefined
513     \def\texosquerycurrencypound{pound}
514   \else
515     \def\texosquerycurrencypound{\pounds}
516   \fi
517 \else
518   \def\texosquerycurrencypound{\faGbp}
519 \fi
```

`\texosquerycurrencysign`

```
520 \ifx\textcurrency\undefined
521   \def\texosquerycurrencysign{currency-sign}
522 \else
523   \def\texosquerycurrencysign{\textcurrency}
524 \fi
```

`\texosquerycurrencyyen`

```
525 \ifx\faYen\undefined
526   \ifx\textyen\undefined
527     \def\texosquerycurrencyyen{yen}
528   \else
529     \def\texosquerycurrencyyen{\textyen}
530   \fi
531 \else
532   \def\texosquerycurrencyyen{\faYen}
533 \fi
```

`\texosquerycurrencyecu`

```
534 \def\texosquerycurrencyecu{ecu}
```

`\texosquerycurrencycolon`

```
535 \def\texosquerycurrencycolon{colon}
```

`\texosquerycurrencycruzeiro`

```
536 \def\texosquerycurrencycruzeiro{cruzeiro}
```

`\texosquerycurrencyfranc`

```
537 \def\texosquerycurrencyfranc{franc}
```

`\texosquerycurrencylira`

```
538 \ifx\textlira\undefined
539   \def\texosquerycurrencylira{lira}
540 \else
541   \def\texosquerycurrencylira{\textlira}
542 \fi
```

`\texosquerycurrencymill`

```
543 \def\texosquerycurrencymill{mill}
```

`\texosquerycurrencynaira`

```
544 \ifx\textnaira\undefined
545   \def\texosquerycurrencynaira{naira}
546 \else
547   \def\texosquerycurrencynaira{\textnaira}
548 \fi
```

`\texosquerycurrencypeseta`

```
549 \def\texosquerycurrencypeseta{peseta}
```

`\texosquerycurrencyrupee`

```
550 \ifx\faRupee\undefined
551   \def\texosquerycurrencyrupee{rupee}
552 \else
553   \def\texosquerycurrencyrupee{\faRupee}
554 \fi
```

`\texosquerycurrencywon`

```
555 \ifx\faWon\undefined
556   \ifx\textwon\undefined
557     \def\texosquerycurrencywon{won}
558   \else
559     \def\texosquerycurrencywon{\textwon}
560   \fi
561 \else
562   \def\texosquerycurrencywon{\faWon}
563 \fi
```

`\texosquerycurrencynewsheqel`

```
564 \ifx\faSheqel\undefined
565   \def\texosquerycurrencynewsheqel{newsheqel}
566 \else
567   \def\texosquerycurrencynewsheqel{\faSheqel}
568 \fi
```

`\texosquerycurrencydong`

```
569 \ifx\textdong\undefined
570   \def\texosquerycurrencydong{dong}
571 \else
572   \def\texosquerycurrencydong{\textdong}
573 \fi
```

`\texosquerycurrencyeuro`

```
574 \ifx\faEuro\undefined
575   \ifx\texteuro\undefined
```

```
576    \ifx\euro\undefined
577      \def\texosquerycurrencyeuro{euro}
578    \else
579      \def\texosquerycurrencyeuro{\euro}
580    \fi
581  \else
582    \def\texosquerycurrencyeuro{\texteuro}
583  \fi
584 \else
585  \def\texosquerycurrencyeuro{\faEuro}
586 \fi
```

\texosquerycurrencykip

```
587 \def\texosquerycurrencykip{kip}
```

\texosquerycurrencytugrik

```
588 \def\texosquerycurrencytugrik{tugrik}
```

\texosquerycurrencydrachma

```
589 \def\texosquerycurrencydrachma{drachma}
```

\texosquerycurrencygermanpenny

```
590 \def\texosquerycurrencygermanpenny{german-penny}
```

\texosquerycurrencypeso

```
591 \ifx\textpeso\undefined
592   \def\texosquerycurrencypeso{peso}
593 \else
594   \def\texosquerycurrencypeso{\textpeso}
595 \fi
```

\texosquerycurrencyguarani

```
596 \ifx\textguarani\undefined
597   \def\texosquerycurrencyguarani{guarani}
598 \else
599   \def\texosquerycurrencyguarani{\textguarani}
600 \fi
```

\texosquerycurrencyaustral

```
601 \def\texosquerycurrencyaustral{austral}
```

\texosquerycurrencyhryvnia

```
602 \def\texosquerycurrencyhryvnia{hryvnia}
```

\texosquerycurrencycedi

```
603 \ifx\textcolonmonetary\undefined
604   \def\texosquerycurrencycedi{cedi}
605 \else
606   \def\texosquerycurrencycedi{\textcolonmonetary}
607 \fi
```

608 `\def\texosquerycurrencylivretournois{livre-tournois}`

\texosquerycurrencyspesmilo

609 `\def\texosquerycurrencyspesmilo{spesmilo}`

\texosquerycurrencytenge

610 `\def\texosquerycurrencytenge{tenge}`

\texosquerycurrencyrupee

611 `\def\texosquerycurrencyrupee{rupee}`

\texosquerycurrencyturkishlira

612 `\ifx\faTurkishLira\undefined`
613 `  \def\texosquerycurrencyturkishlira{turkish-lira}`
614 `\else`
615 `  \def\texosquerycurrencyturkishlira{\faTurkishLira}`
616 `\fi`

\texosquerycurrencynordicmark

617 `\def\texosquerycurrencynordicmark{nordic-mark}`

\texosquerycurrencymanat

618 `\def\texosquerycurrencymanat{manat}`

\texosquerycurrencyruble

619 `\ifx\faRuble\undefined`
620 `  \def\texosquerycurrencyruble{ruble}`
621 `\else`
622 `  \def\texosquerycurrencyruble{\faRuble}`
623 `\fi`

3.1.2 Shortcut Commands

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.

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

\TeXOSQueryLangTag Query the language tag and store the result in the control sequence provided in the argument.

625 `\def\TeXOSQueryLangTag#1{\TeXOSQuery{#1}{\string-b}}`

\TeXOSQueryNumeric Query the numeric settings for the locale given in the second argument and store the result in the control sequence provided in the argument. Leave the second argument empty for the default locale.

626 `\def\TeXOSQueryNumeric#1#2{\TeXOSQuery{#1}{\string-N #2}}`

\TeXOSQueryLocaleData	Query the data for the locale given in the second argument and store the result in the control sequence provided in the argument. Leave the second argument empty for the default locale.

```
627 \def\TeXOSQueryLocaleData#1#2{\TeXOSQuery{#1}{\string-D #2}}
```

\TeXOSQueryCwd	Query the current working directory.

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

\TeXOSQueryHome	Query the user's home directory.

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

\TeXOSQueryTmpDir	Query the temporary directory.

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

\TeXOSQueryVersion	Query the operating system version.

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

\TeXOSQueryArch	Query the operating system architecture.

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

\TeXOSQueryName	Query the operating system name.

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

\TeXOSQueryDateTime	Query the current date and time.

```
634 \def\TeXOSQueryDateTime#1{%
635   \TeXOSQuery{#1}{\string-M}%
636 }
```

\TeXOSQueryTimeZones	Query the current time zone mappings. Leave the second argument empty for the default locale.

```
637 \def\TeXOSQueryTimeZones#1#2{%
638   \TeXOSQuery{#1}{\string-Z #2}%
639 }
```

\TeXOSQueryNow	Query the current time stamp.

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

The D needs category code 12 just in case texosquery is running in a backward compatibility mode that doesn't use \pdfd.

```
641   \edef\@texosquery@restore@D{%
642     \noexpand\catcode`\noexpand\D=\the\catcode`\D\relax}%
643   \catcode`\D=12\relax
644   \TeXOSQuery{#1}{\string-n}%
645   \@texosquery@restore@D
646 }
```

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

`\texosquerystripquotes`

```
647 \def\texosquerystripquotes#1{%
648   \@texosquery@stripquotes#1\@mid@texosquery@stripquotes
649     "\relax"\relax\@end@texosquery@stripquotes
650 }
651 \def\@texosquery@stripquotes#1"#2"{%
652   \@@texosquery@stripquotes#1#2%
653 }
654 \def\@@texosquery@stripquotes#1\@mid@texosquery@stripquotes#2\@end@texosquery@stripquotes{%
655   #1%
656 }
```

The restricted mode doesn't permit quotes in the shell escape, so arguments are only quoted in unrestricted mode.

`\@texosquery@argquote`

```
657 \ifnum\texosquery@shellescape=2\relax
658   \def\@texosquery@argquote#1{#1}
659 \else
660   \def\@texosquery@argquote#1{\string'#1\string'}
661 \fi
```

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

```
662 \def\TeXOSQueryFileDate#1#2{%
```

The D needs category code 12 just in case texosquery is running in a backward compatibility mode that doesn't use `\pdfd`.

```
663   \edef\@texosquery@restore@D{%
664     \noexpand\catcode`\noexpand\D=\the\catcode`\D\relax}%
665   \catcode`\D=12\relax
666   \TeXOSQuery{#1}{\string-d
667     \@texosquery@argquote{\texosquerystripquotes{#2}}}%
668   \@texosquery@restore@D
669 }
```

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

```
670 \def\TeXOSQueryFileSize#1#2{\TeXOSQuery{#1}{\string-s
671 \@texosquery@argquote{\texosquerystripquotes{#2}}}}
```

`\@texosquery@filelist`

```
672 \def\@texosquery@filelist#1#2#3#4#5{\TeXOSQuery{#1}{%
673 \string#2 \@texosquery@argquote{#3}
674 \@texosquery@argquote{\texosquerystripquotes{#4}} #5}}
```

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

```
675 \def\TeXOSQueryFileList#1#2#3{%
676 \@texosquery@filelist{#1}{-i}{#2}{#3}{}%
677 }
```

`\TeXOSQueryFileListDateAsc`   As above, but sort by date.

```
678 \def\TeXOSQueryFileListDateAsc#1#2#3{%
679 \@texosquery@filelist{#1}{-i}{#2}{#3}{date}%
680 }
```

`\TeXOSQueryFileListDateDes`   As above, but sort by date in descending order.

```
681 \def\TeXOSQueryFileListDateDes#1#2#3{%
682 \@texosquery@filelist{#1}{-i}{#2}{#3}{date\string-des}%
683 }
```

`\TeXOSQueryFileListSizeAsc`   As above, but sort by size.

```
684 \def\TeXOSQueryFileListSizeAsc#1#2#3{%
685 \@texosquery@filelist{#1}{-i}{#2}{#3}{size}%
686 }
```

`\TeXOSQueryFileListSizeDes`   As above, but sort by size in descending order.

```
687 \def\TeXOSQueryFileListSizeDes#1#2#3{%
688 \@texosquery@filelist{#1}{-i}{#2}{#3}{size\string-des}%
689 }
```

`\TeXOSQueryFileListNameAsc`   As above, but sort by name.

```
690 \def\TeXOSQueryFileListNameAsc#1#2#3{%
691 \@texosquery@filelist{#1}{-i}{#2}{#3}{name}%
692 }
```

`\TeXOSQueryFileListNameDes`   As above, but sort by name in descending order.

```
693 \def\TeXOSQueryFileListNameDes#1#2#3{%
694 \@texosquery@filelist{#1}{-i}{#2}{#3}{name\string-des}%
695 }
```

`SQueryFileListNameIgnoreCaseAsc`   As above, but sort by case-insensitive name.

```
696 \def\TeXOSQueryFileListNameIgnoreCaseAsc#1#2#3{%
697 \@texosquery@filelist{#1}{-i}{#2}{#3}{iname}%
698 }
```

`SQueryFileListNameIgnoreCaseDes`   As above, but sort by case-insensitive name in descending order.

```
699 \def\TeXOSQueryFileListNameIgnoreCaseDes#1#2#3{%
700 \@texosquery@filelist{#1}{-i}{#2}{#3}{iname\string-des}%
701 }
```

`\TeXOSQueryFileListExtAsc`   As above, but sort by extension.

```
702 \def\TeXOSQueryFileListExtAsc#1#2#3{%
703 \@texosquery@filelist{#1}{-i}{#2}{#3}{ext}%
704 }
```

`\TeXOSQueryFileListExtDes`   As above, but sort by extension in descending order.

```
705 \def\TeXOSQueryFileListExtDes#1#2#3{%
706 \@texosquery@filelist{#1}{-i}{#2}{#3}{ext\string-des}%
707 }
```

66

| `\TeXOSQueryRegularFileList` | List all regular files. |

```
708 \def\TeXOSQueryRegularFileList#1#2#3{%
709 \@texosquery@filelist{#1}{-ir}{#2}{#3}{}%
710 }
```

| `\TeXOSQuerySubDirList` | List all sub-directories. |

```
711 \def\TeXOSQuerySubDirList#1#2#3{%
712 \@texosquery@filelist{#1}{-id}{#2}{#3}{}%
713 }
```

| `\TeXOSQueryRegularFileListDateAsc` | List all regular files sorted by date. |

```
714 \def\TeXOSQueryRegularFileListDateAsc#1#2#3{%
715 \@texosquery@filelist{#1}{-ir}{#2}{#3}{date}%
716 }
```

| `\TeXOSQuerySubDirListDateAsc` | List all sub-directories sorted by date. |

```
717 \def\TeXOSQuerySubDirListDateAsc#1#2#3{%
718 \@texosquery@filelist{#1}{-id}{#2}{#3}{date}%
719 }
```

| `\TeXOSQueryRegularFileListDateDes` | List all regular files sorted by date in descending order. |

```
720 \def\TeXOSQueryRegularFileListDateDes#1#2#3{%
721 \@texosquery@filelist{#1}{-ir}{#2}{#3}{date\string-des}%
722 }
```

| `\TeXOSQuerySubDirListDateDes` | List all sub-directories sorted by date in descending order. |

```
723 \def\TeXOSQuerySubDirListDateDes#1#2#3{%
724 \@texosquery@filelist{#1}{-id}{#2}{#3}{date\string-des}%
725 }
```

| `\TeXOSQueryRegularFileListSizeAsc` | List all regular files sorted by size. |

```
726 \def\TeXOSQueryRegularFileListSizeAsc#1#2#3{%
727 \@texosquery@filelist{#1}{-ir}{#2}{#3}{size}%
728 }
```

| `\TeXOSQuerySubDirListSizeAsc` | List all sub-directories sorted by size. |

```
729 \def\TeXOSQuerySubDirListSizeAsc#1#2#3{%
730 \@texosquery@filelist{#1}{-id}{#2}{#3}{size}%
731 }
```

| `\TeXOSQueryRegularFileListSizeDes` | List all regular files sorted by size in descending order. |

```
732 \def\TeXOSQueryRegularFileListSizeDes#1#2#3{%
733 \@texosquery@filelist{#1}{-ir}{#2}{#3}{size\string-des}%
734 }
```

| `\TeXOSQuerySubDirListSizeDes` | List all sub-directories sorted by size in descending order. |

```
735 \def\TeXOSQuerySubDirListSizeDes#1#2#3{%
736 \@texosquery@filelist{#1}{-id}{#2}{#3}{size\string-des}%
737 }
```

`\TeXOSQueryRegularFileListNameAsc` List all regular files sorted by file name.

```
738 \def\TeXOSQueryRegularFileListNameAsc#1#2#3{%
739 \@texosquery@filelist{#1}{-ir}{#2}{#3}{name}%
740 }
```

`\TeXOSQuerySubDirListNameAsc` List all sub-directories sorted by file name.

```
741 \def\TeXOSQuerySubDirListNameAsc#1#2#3{%
742 \@texosquery@filelist{#1}{-id}{#2}{#3}{name}%
743 }
```

`\TeXOSQueryRegularFileListNameDes` List all regular files sorted by file name in descending order.

```
744 \def\TeXOSQueryRegularFileListNameDes#1#2#3{%
745 \@texosquery@filelist{#1}{-ir}{#2}{#3}{name\string-des}%
746 }
```

`\TeXOSQuerySubDirListNameDes` List all sub-directories sorted by name in descending order.

```
747 \def\TeXOSQuerySubDirListNameDes#1#2#3{%
748 \@texosquery@filelist{#1}{-id}{#2}{#3}{name\string-des}%
749 }
```

`\TeXOSQueryRegularFileListNameIgnoreCaseAsc` List all regular files sorted by file case-insensitive name.

```
750 \def\TeXOSQueryRegularFileListNameIgnoreCaseAsc#1#2#3{%
751 \@texosquery@filelist{#1}{-ir}{#2}{#3}{iname}%
752 }
```

`\TeXOSQuerySubDirListNameIgnoreCaseAsc` List all sub-directories sorted by file case-insensitive name.

```
753 \def\TeXOSQuerySubDirListNameIgnoreCaseAsc#1#2#3{%
754 \@texosquery@filelist{#1}{-id}{#2}{#3}{iname}%
755 }
```

`\TeXOSQueryRegularFileListNameIgnoreCaseDes` List all regular files sorted by file case-insensitive name in descending order.

```
756 \def\TeXOSQueryRegularFileListNameIgnoreCaseDes#1#2#3{%
757 \@texosquery@filelist{#1}{-ir}{#2}{#3}{iname\string-des}%
758 }
```

`\TeXOSQuerySubDirListNameIgnoreCaseDes` List all sub-directories sorted by case-insensitive name in descending order.

```
759 \def\TeXOSQuerySubDirListNameIgnoreCaseDes#1#2#3{%
760 \@texosquery@filelist{#1}{-id}{#2}{#3}{iname\string-des}%
761 }
```

`\TeXOSQueryRegularFileListExtAsc` List all regular files sorted by file extension.

```
762 \def\TeXOSQueryRegularFileListExtAsc#1#2#3{%
763 \@texosquery@filelist{#1}{-ir}{#2}{#3}{ext}%
764 }
```

`\TeXOSQuerySubDirListExtAsc` List all sub-directories sorted by file extension.

```
765 \def\TeXOSQuerySubDirListExtAsc#1#2#3{%
766 \@texosquery@filelist{#1}{-id}{#2}{#3}{ext}%
767 }
```

\TeXOSQueryRegularFileListExtDes List all regular files sorted by file extension in descending order.

```
768 \def\TeXOSQueryRegularFileListExtDes#1#2#3{%
769 \@texosquery@filelist{#1}{-ir}{#2}{#3}{ext\string-des}%
770 }
```

\TeXOSQuerySubDirListExtDes List all sub-directories sorted by extension in descending order.

```
771 \def\TeXOSQuerySubDirListExtDes#1#2#3{%
772 \@texosquery@filelist{#1}{-id}{#2}{#3}{ext\string-des}%
773 }
```

\@texosquery@filterfilelist

```
774 \def\@texosquery@filterfilelist#1#2#3#4#5#6{%
775 \TeXOSQuery{#1}%
776 {%
777     \string#2
778     \@texosquery@argquote{#3}
779     \@texosquery@argquote{#4}
780     \@texosquery@argquote{\texosquerystripquotes{#5}} #6%
781 }%
782 }
```

\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!*

```
783 \def\TeXOSQueryFilterFileList#1#2#3#4{%
784 \@texosquery@filterfilelist{#1}{-f}{#2}{#3}{#4}{}%
785 }
```

\TeXOSQueryFilterFileListDateAsc As above, but sort by date.

```
786 \def\TeXOSQueryFilterFileListDateAsc#1#2#3#4{%
787 \@texosquery@filterfilelist{#1}{-f}{#2}{#3}{#4}{date}%
788 }
```

\TeXOSQueryFilterFileListDateDes As above, but sort by date in descending order.

```
789 \def\TeXOSQueryFilterFileListDateDes#1#2#3#4{%
790 \@texosquery@filterfilelist{#1}{-f}{#2}{#3}{#4}{date\string-des}%
791 }
```

\TeXOSQueryFilterFileListSizeAsc As above, but sort by size.

```
792 \def\TeXOSQueryFilterFileListSizeAsc#1#2#3#4{%
793 \@texosquery@filterfilelist{#1}{-f}{#2}{#3}{#4}{size}%
794 }
```

\TeXOSQueryFilterFileListSizeDes As above, but sort by size in descending order.

```
795 \def\TeXOSQueryFilterFileListSizeDes#1#2#3#4{%
796 \@texosquery@filterfilelist{#1}{-f}{#2}{#3}{#4}{size\string-des}%
797 }
```

TeXOSQueryFilterFileListNameAsc	As above, but sort by file name.

```
798 \def\TeXOSQueryFilterFileListNameAsc#1#2#3#4{%
799 \@texosquery@filterfilelist{#1}{-f}{#2}{#3}{#4}{name}%
800 }
```

TeXOSQueryFilterFileListNameDes	As above, but sort by name in descending order.

```
801 \def\TeXOSQueryFilterFileListNameDes#1#2#3#4{%
802 \@texosquery@filterfilelist{#1}{-f}{#2}{#3}{#4}{name\string-des}%
803 }
```

FilterFileListNameIgnoreCaseAsc	As above, but sort by file name (case-insensitive).

```
804 \def\TeXOSQueryFilterFileListNameIgnoreCaseAsc#1#2#3#4{%
805 \@texosquery@filterfilelist{#1}{-f}{#2}{#3}{#4}{iname}%
806 }
```

FilterFileListNameIgnoreCaseDes	As above, but sort by name in descending order (case-insensitive).

```
807 \def\TeXOSQueryFilterFileListNameIgnoreCaseDes#1#2#3#4{%
808 \@texosquery@filterfilelist{#1}{-f}{#2}{#3}{#4}{iname\string-des}%
809 }
```

\TeXOSQueryFilterFileListExtAsc	As above, but sort by file extension.

```
810 \def\TeXOSQueryFilterFileListExtAsc#1#2#3#4{%
811 \@texosquery@filterfilelist{#1}{-f}{#2}{#3}{#4}{ext}%
812 }
```

\TeXOSQueryFilterFileListExtDes	As above, but sort by extension in descending order.

```
813 \def\TeXOSQueryFilterFileListExtDes#1#2#3#4{%
814 \@texosquery@filterfilelist{#1}{-f}{#2}{#3}{#4}{ext\string-des}%
815 }
```

TeXOSQueryFilterRegularFileList	Filtered list or regular files.

```
816 \def\TeXOSQueryFilterRegularFileList#1#2#3#4{%
817 \@texosquery@filterfilelist{#1}{-fr}{#2}{#3}{#4}{}%
818 }
```

\TeXOSQueryFilterSubDirList	Filtered list of sub-directories.

```
819 \def\TeXOSQueryFilterSubDirList#1#2#3#4{%
820 \@texosquery@filterfilelist{#1}{-fd}{#2}{#3}{#4}{}%
821 }
```

XOSQueryFilterSubDirListDateAsc	Filtered sort of sub-directories by file date.

```
822 \def\TeXOSQueryFilterSubDirListDateAsc#1#2#3#4{%
823 \@texosquery@filterfilelist{#1}{-fd}{#2}{#3}{#4}{date}%
824 }
```

eryFilterRegularFileListDateAsc	Filtered sort of regular files by file date.

```
825 \def\TeXOSQueryFilterRegularFileListDateAsc#1#2#3#4{%
826 \@texosquery@filterfilelist{#1}{-fr}{#2}{#3}{#4}{date}%
827 }
```

XOSQueryFilterSubDirListDateDes	Filtered sort of sub-directories by file date in descending order.

```
828 \def\TeXOSQueryFilterSubDirListDateDes#1#2#3#4{%
829 \@texosquery@filterfilelist{#1}{-fd}{#2}{#3}{#4}{date\string-des}%
830 }
```

eryFilterRegularFileListDateDes	Filtered sort of regular files by file date in descending order.

```
831 \def\TeXOSQueryFilterRegularFileListDateDes#1#2#3#4{%
832 \@texosquery@filterfilelist{#1}{-fr}{#2}{#3}{#4}{date\string-des}%
833 }
```

XOSQueryFilterSubDirListSizeAsc	Filtered sort of sub-directories by file size.

```
834 \def\TeXOSQueryFilterSubDirListSizeAsc#1#2#3#4{%
835 \@texosquery@filterfilelist{#1}{-fd}{#2}{#3}{#4}{size}%
836 }
```

eryFilterRegularFileListSizeAsc	Filtered sort of regular files by file size.

```
837 \def\TeXOSQueryFilterRegularFileListSizeAsc#1#2#3#4{%
838 \@texosquery@filterfilelist{#1}{-fr}{#2}{#3}{#4}{size}%
839 }
```

XOSQueryFilterSubDirListSizeDes	Filtered sort of sub-directories by file size in descending order.

```
840 \def\TeXOSQueryFilterSubDirListSizeDes#1#2#3#4{%
841 \@texosquery@filterfilelist{#1}{-fd}{#2}{#3}{#4}{size\string-des}%
842 }
```

eryFilterRegularFileListSizeDes	Filtered sort of regular files by file size in descending order.

```
843 \def\TeXOSQueryFilterRegularFileListSizeDes#1#2#3#4{%
844 \@texosquery@filterfilelist{#1}{-fr}{#2}{#3}{#4}{size\string-des}%
845 }
```

XOSQueryFilterSubDirListNameAsc	Filtered sort of sub-directories by file name.

```
846 \def\TeXOSQueryFilterSubDirListNameAsc#1#2#3#4{%
847 \@texosquery@filterfilelist{#1}{-fd}{#2}{#3}{#4}{name}%
848 }
```

eryFilterRegularFileListNameAsc	Filtered sort of regular files by file name.

```
849 \def\TeXOSQueryFilterRegularFileListNameAsc#1#2#3#4{%
850 \@texosquery@filterfilelist{#1}{-fr}{#2}{#3}{#4}{name}%
851 }
```

XOSQueryFilterSubDirListNameDes	Filtered sort of sub-directories by file name in descending order.

```
852 \def\TeXOSQueryFilterSubDirListNameDes#1#2#3#4{%
853 \@texosquery@filterfilelist{#1}{-fd}{#2}{#3}{#4}{name\string-des}%
854 }
```

eryFilterRegularFileListNameDes	Filtered sort of regular files by file name in descending order.

```
855 \def\TeXOSQueryFilterRegularFileListNameDes#1#2#3#4{%
856 \@texosquery@filterfilelist{#1}{-fr}{#2}{#3}{#4}{name\string-des}%
857 }
```

...lterSubDirListNameIgnoreCaseAsc	Filtered sort of sub-directories by case-insensitive file name.

```
858 \def\TeXOSQueryFilterSubDirListNameIgnoreCaseAsc#1#2#3#4{%
859 \@texosquery@filterfilelist{#1}{-fd}{#2}{#3}{#4}{iname}%
860 }
```

...egularFileListNameIgnoreCaseAsc	Filtered sort of regular files by case-insensitive file name.

```
861 \def\TeXOSQueryFilterRegularFileListNameIgnoreCaseAsc#1#2#3#4{%
862 \@texosquery@filterfilelist{#1}{-fr}{#2}{#3}{#4}{iname}%
863 }
```

...lterSubDirListNameIgnoreCaseDes	Filtered sort of sub-directories by case-insensitive file name in descending order.

```
864 \def\TeXOSQueryFilterSubDirListNameIgnoreCaseDes#1#2#3#4{%
865 \@texosquery@filterfilelist{#1}{-fd}{#2}{#3}{#4}{iname\string-des}%
866 }
```

...egularFileListNameIgnoreCaseDes	Filtered sort of regular files by case-insensitive file name in descending order.

```
867 \def\TeXOSQueryFilterRegularFileListNameIgnoreCaseDes#1#2#3#4{%
868 \@texosquery@filterfilelist{#1}{-fr}{#2}{#3}{#4}{iname\string-des}%
869 }
```

...eXOSQueryFilterSubDirListExtAsc	Filtered sort of sub-directories by file extension. (Added for completeness as directories don't tend to have extensions.)

```
870 \def\TeXOSQueryFilterSubDirListExtAsc#1#2#3#4{%
871 \@texosquery@filterfilelist{#1}{-fd}{#2}{#3}{#4}{ext}%
872 }
```

...ueryFilterRegularFileListExtAsc	Filtered sort of regular files by file extension.

```
873 \def\TeXOSQueryFilterRegularFileListExtAsc#1#2#3#4{%
874 \@texosquery@filterfilelist{#1}{-fd}{#2}{#3}{#4}{ext}%
875 }
```

...eXOSQueryFilterSubDirListExtDes	Filtered sort of sub-directories by file extension in descending order.

```
876 \def\TeXOSQueryFilterSubDirListExtDes#1#2#3#4{%
877 \@texosquery@filterfilelist{#1}{-fd}{#2}{#3}{#4}{ext\string-des}%
878 }
```

...ueryFilterRegularFileListExtDes	Filtered sort of regular files by file extension in descending order.

```
879 \def\TeXOSQueryFilterRegularFileListExtDes#1#2#3#4{%
880 \@texosquery@filterfilelist{#1}{-fr}{#2}{#3}{#4}{ext\string-des}%
881 }
```

\@texosquery@walk

```
882 \def\@texosquery@walk#1#2#3#4#5{%
883 \TeXOSQuery{#1}%
884 {%
885    \string-w
886    \@texosquery@argquote{#2}
887    \@texosquery@argquote{#3}
```

```
888     \@texosquery@argquote{\texosquerystripquotes{#4}} #5%
889 }%
890 }
```

\TeXOSQueryWalk Recursive filtered listing of regular files.

```
891 \def\TeXOSQueryWalk#1#2#3#4{%
892 \@texosquery@walk{#1}{#2}{#3}{#4}{}%
893 }
```

\TeXOSQueryWalkDateAsc As above, but sort by date.

```
894 \def\TeXOSQueryWalkDateAsc#1#2#3#4{%
895 \@texosquery@walk{#1}{#2}{#3}{#4}{date}%
896 }
```

\TeXOSQueryWalkDateDes As above, but sort by date in descending order.

```
897 \def\TeXOSQueryWalkDateDes#1#2#3#4{%
898 \@texosquery@walk{#1}{#2}{#3}{#4}{date\string-des}%
899 }
```

\TeXOSQueryWalkSizeAsc As above, but sort by file size.

```
900 \def\TeXOSQueryWalkSizeAsc#1#2#3#4{%
901 \@texosquery@walk{#1}{#2}{#3}{#4}{size}%
902 }
```

\TeXOSQueryWalkSizeDes As above, but sort by file size in descending order.

```
903 \def\TeXOSQueryWalkSizeDes#1#2#3#4{%
904 \@texosquery@walk{#1}{#2}{#3}{#4}{size\string-des}%
905 }
```

\TeXOSQueryWalkNameAsc As above, but sort by file name.

```
906 \def\TeXOSQueryWalkNameAsc#1#2#3#4{%
907 \@texosquery@walk{#1}{#2}{#3}{#4}{name}%
908 }
```

\TeXOSQueryWalkNameDes As above, but sort by file name in descending order.

```
909 \def\TeXOSQueryWalkNameDes#1#2#3#4{%
910 \@texosquery@walk{#1}{#2}{#3}{#4}{name\string-des}%
911 }
```

\TeXOSQueryWalkNameIgnoreCaseAsc As above, but sort by file name (case-insensitive).

```
912 \def\TeXOSQueryWalkNameIgnoreCaseAsc#1#2#3#4{%
913 \@texosquery@walk{#1}{#2}{#3}{#4}{iname}%
914 }
```

\TeXOSQueryWalkNameIgnoreCaseDes As above, but sort by file name (case-insensitive) in descending order.

```
915 \def\TeXOSQueryWalkNameIgnoreCaseDes#1#2#3#4{%
916 \@texosquery@walk{#1}{#2}{#3}{#4}{iname\string-des}%
917 }
```

\TeXOSQueryWalkExtAsc	As above, but sort by file extension.

```
918 \def\TeXOSQueryWalkExtAsc#1#2#3#4{%
919 \@texosquery@walk{#1}{#2}{#3}{#4}{ext}%
920 }
```

\TeXOSQueryWalkExtDes	As above, but sort by file extension in descending order.

```
921 \def\TeXOSQueryWalkExtDes#1#2#3#4{%
922 \@texosquery@walk{#1}{#2}{#3}{#4}{ext\string-des}%
923 }
```

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

```
924 \def\TeXOSQueryFileURI#1#2{\TeXOSQuery{#1}{\string-u
925 \@texosquery@argquote{\texosquerystripquotes{#2}}}}
```

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

```
926 \def\TeXOSQueryFilePath#1#2{\TeXOSQuery{#1}{\string-p
927 \@texosquery@argquote{\texosquerystripquotes{#2}}}}
```

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

```
928 \def\TeXOSQueryDirName#1#2{\TeXOSQuery{#1}{\string-e
929 \@texosquery@argquote{\texosquerystripquotes{#2}}}}
```

3.1.3 Pattern Formats

There are two basic types of patterns: date/time or numeric. A pattern is stored in a control sequence using custom markup that's easier for TeX to parse than it would be to parse strings in the form YYYY-MM or #,##0. This internal pattern format can be obtained through capturing the output of texosquery's -D action, but patterns can also be constructed using

\texosquerydefpattern{⟨cs⟩}{⟨pattern specs⟩}

The pattern is stored in ⟨cs⟩. The ⟨pattern specs⟩ depend on whether a date-time or numeric pattern is required. For a date-time pattern, each date/time element is identified using

\texosquerydtf{⟨n⟩}{⟨identifier⟩}

where ⟨identifier⟩ identifies the element type (such as M for month or s for seconds) and ⟨n⟩ indicates how the element should be formatted, where ⟨n⟩ is an integer from 1 to 4. For example if ⟨n⟩ is 2 and ⟨identifier⟩ is M, then this indicates the MM format, which produces a two-digit number.

Since it's rather cumbersome to keep typing \texosquerydtf and it can make for rather hard to read code, \texosquerydefpattern locally redefines \% to expand to \texosquerydtf. This means that if you do, for example:

\texosquerydefpattern{\pattern}{\%2d/\%2M/\%4y}

then \pattern is defined to

\texosquerydtf 2d/\texosquerydtf 2M/\texosquerydtf 4y

74

When simply used within the document, this just expands to the pattern format. For example:

```
Pattern: \pattern.
```

will display "Pattern: dd/MM/yyyy" in the PDF. However, when used with \texosqueryfmtdatetime, the definition of \texosquerydtf changes to reproduce the required date/time element.

For example:

```
\texosquerydefpattern{\pattern}{\%2d/\%2M/\%4y \%2H:\%2m:\%2s \%2Z}

Pattern: \pattern.

\TeXOSQueryDateTime{\datetimedata}

\ifx\datetimedata\empty
 Query Failed!
\else
 \expandafter\texosqueryfmtdatetime\expandafter\pattern\datetimedata
\fi
```

The numeric patterns are rather more complicated. The ⟨*pattern specs*⟩ now needs to use the following formats:

```
\texosquerypatnum{⟨+ve/-ve numeric pattern⟩}
```

This is a numeric pattern applied to a number regardless of whether the number is positive or negative. (If negative, the minus sign is automatically inserted.) This is rather a long and cumbersome command to type, so \texosquerydefpattern locally defines \numfmt to expand to it.

```
\texosquerypatplusminus{⟨+ve numeric pattern⟩}{⟨-ve numeric pattern⟩}
```

This provides a pattern ⟨*+ve numeric pattern*⟩ to use if the number is positive and a patter ⟨*-ve numeric pattern*⟩ to use if the number is negative. Again \texosquerydefpattern locally defines a shortcut, \pmnumfmt, to expand to this command.

```
\texosquerypatsinum{⟨decimal pattern⟩}{⟨mantissa pattern⟩}
```

This provides a pattern to use for SI numbers where ⟨*decimal pattern*⟩ is a pattern for the decimal number part (before the exponent symbol) and ⟨*mantissa*⟩ is the pattern for the integer part in the mantissa (after the exponent symbol). The locally defined shortcut is \sinumfmt. The ⟨*decimal pattern*⟩ will typically be in the form:

```
\texosquerypatdec{⟨integer pattern⟩}{⟨fraction pattern⟩}
```

This indicates a decimal pattern where the ⟨*integer pattern*⟩ is applied to the part before the decimal separator and ⟨*fraction pattern*⟩ is applied to the part after the separator. The shortcut is \decfmt.

```
\texosquerypatprefixcurrency{⟨decimal pattern⟩}{⟨text⟩}
```

This indicates a currency pattern with a prefixed currency symbol where ⟨*text*⟩ is inserted before the currency symbol. The shortcut is \pcur. The ⟨*decimal pattern*⟩ will typically use \texosquerypatdec{⟨*int*⟩}⟨*frac*⟩. (Similarly for the following.)

\texosquerypatprefixicurrency{⟨*decimal pattern*⟩}{⟨*text*⟩}

This indicates an international currency pattern with a prefixed international currency symbol where ⟨*text*⟩ is inserted before the symbol. The shortcut is \picur.

\texosquerypatsuffixcurrency{⟨*decimal pattern*⟩}{⟨*text*⟩}

This indicates a currency pattern with a suffixed international currency symbol where ⟨*text*⟩ is inserted after the currency symbol. The shortcut is \scur.

\texosquerypatsuffixicurrency{⟨*decimal pattern*⟩}{⟨*text*⟩}

This indicates an international currency pattern with a suffixed international currency symbol where ⟨*text*⟩ is inserted after the symbol. The shortcut is \sicur.

\texosquerypatprefixpercent{⟨*decimal pattern*⟩}{⟨*text*⟩}

This indicates a percentage pattern with a prefixed percent symbol where ⟨*text*⟩ is inserted before the symbol. The shortcut is \ppct.

\texosquerypatsuffixpercent{⟨*decimal pattern*⟩}{⟨*text*⟩}

This indicates a percentage pattern with a suffixed percent symbol where ⟨*text*⟩ is inserted after the symbol. The shortcut is \spct.

\texosquerypatprefixpermill{⟨*decimal pattern*⟩}{⟨*text*⟩}

This indicates a per-mill pattern with a prefixed per-mill symbol where ⟨*text*⟩ is inserted before the symbol. The shortcut is \ppml.

\texosquerypatsuffixpermill{⟨*decimal pattern*⟩}{⟨*text*⟩}

This indicates a per-mill pattern with a suffixed per-mill symbol where ⟨*text*⟩ is inserted after the symbol. The shortcut is \spml.

> **Important Note:** The integer parts ⟨*integer part*⟩, ⟨*fraction part*⟩ and ⟨*mantissa*⟩ must have *exactly* ten digit identifiers. (TEX can't reach 11 digit numbers.)

There are two types of digit identifiers:

\texosquerypatdigit

This indicates a digit that must be displayed, even if it's not significant (for example a leading zero). The shortcut command is \0 (backslash zero).

```
\texosquerypatdigitnozero
```

This indicates a digit that should only be displayed if it's significant. (For example, if it's a leading zero, it's not shown.) The shortcut command is \# (backslash hash).

   The number group separator can be inserted using

```
\texosquerypatgroupsep
```

The shortcut command is \, (backslash comma).

   The sign can be inserted using

```
\texosquerypatminus
```

This ensures the sign is displayed even if the number is positive. The shortcut command is \- (backslash hyphen).

   Here's an example of a decimal pattern:

```
\texosquerydefpattern{\numpattern}{%
 \numfmt{\decfmt{\#\,\#\#\#\,\#\#\#\,\#\#\0}{\0\#\#\#\#\#\#\#\#}}}
```

The pattern can be applied to a number using \texosqueryfmtnumber:

```
\texosqueryfmtnumber{\numpattern}{123}{4567}{2}
```

which produces: 12,345.67 (the group and decimal separators can be redefined as appropriate).

   Here's an example of a scientific number:

```
\texosquerydefpattern{\sinumpattern}{%
 \sinumfmt
 {\decfmt{\#\,\#\#\#\,\#\#\#\,\#\#\0}{\0\#\#\#\#\#\#\#\#}}%
 {\-\#\#\#\#\#\#\#\0\0}%
}
```

The pattern can be applied to a number:

```
\texosqueryfmtnumber{\sinumpattern}{1}{234567}{3}
```

which produces: 1.234567E+03

   Here's an integer pattern:

```
\texosquerydefpattern{\intpattern}{%
 \patnumfmt{\#\,\#\#\#\,\#\#\#\,\#\#\0}}
```

The pattern applied to a number:

```
\texosqueryfmtnumber{\intpattern}{123}{4567}{2}
```

which produces: 12,345 (the fractional part has been omitted).

   Here's a currency pattern that applies a different format for positive and negative numbers:

```
\texosquerydefpattern{\curpattern}{%
 \pmnumfmt
 {\pcur{\decfmt{\#\,\#\#\#\,\#\#\#\,\#\#\0}{\0\0\#\#\#\#\#\#\#}}{}}%
 {\pcur{\decfmt{\#\,\#\#\#\,\#\#\#\,\#\#\0}{\0\0\#\#\#\#\#\#\#}}{\-}}}
```

This uses the ⟨*text*⟩ part of \pcur to insert the sign before the currency symbol (but only for negative values).

```
\texosqueryfmtnumber{\curpattern}{-1234567}{0}{0}
```

This produces: −$12,345,678.00 (again the symbol and separators can be redefined as appropriate).

Here's an example of a percentage pattern:

```
\texosquerydefpattern{\pcpattern}{%
 \numfmt{\spct{\#\,\#\#\#\,\#\#\#\,\#\#\0}{}}}
```

The pattern can similarly be applied to a number using \texosqueryfmtnumber.

\texosquerydtf   Date/time format placeholder. The second argument is the placeholder character and the first argument is the number of occurrences of that character in the placeholder. The default definition just converts it back to pattern format used by Java's SimpleDateFormat class. The pattern interprets ≥ 4 as a single case, so this will only produce a maximum of four characters.

```
930 \def\texosquerydtf#1#2{%
931   \ifcase#1
932   \or
933    #2%
934   \or
935    #2#2%
936   \or
937    #2#2#2%
938   \else
939    #2#2#2#2%
940   \fi
941 }
```

The following commands are used to display the pattern in the document text to reproduce the pattern string recognised by Java. This is provided for debugging to check the pattern. In most cases the pattern will be applied to a number rather than simply displayed.

\texosquerypatstr   Quoted string contained in number format.

```
942 \def\texosquerypatstr#1{'#1'}
```

\texosquerypatquote   Literal quote contained in number format.

```
943 \def\texosquerypatquote{''}
```

Number format place holders.

\texosquerypatplusminus

```
944 \def\texosquerypatplusminus#1#2{#1;#2}
```

\texosquerypatnum

```
945 \def\texosquerypatnum#1{#1}
```

<dl>

\texosquerypatsinum

946 \def\texosquerypatsinum#1#2{#1E#2}

\texosquerypatdec

947 \def\texosquerypatdec#1#2{#1.#2}

\texosquerypatprefixcurrency First argument is a number, the second is optional text before the currency symbol. This will require UTF-8 support otherwise it will need redefining as appropriate. (Similarly for the other currency commands and for the per-mill commands.)

948 \def\texosquerypatprefixcurrency#1#2{#2¤#1}

\texosquerypatprefixicurrency As above but use international currency symbol.

949 \def\texosquerypatprefixicurrency#1#2{#2¤¤#1}

\texosquerypatsuffixcurrency First argument is a number, the second is optional text after the currency symbol.

950 \def\texosquerypatsuffixcurrency#1#2{#1¤#2}

\texosquerypatsuffixicurrency As above but use international currency symbol.

951 \def\texosquerypatsuffixicurrency#1#2{#1¤¤#2}

\texosquerypatdigit

952 \def\texosquerypatdigit{0}

\texosquerypatdigitnozero

953 \def\texosquerypatdigitnozero{\#}

\texosquerypatminus

954 \def\texosquerypatminus{-}

\texosquerypatgroupsep

955 \def\texosquerypatgroupsep{,}

\texosquerypatprefixpercent The first argument is the value, the second argument is optional text before the percent symbol.

956 \def\texosquerypatprefixpercent#1#2{#2\%#1}

\texosquerypatsuffixpercent The first argument is the value, the second argument is optional text after the percent symbol.

957 \def\texosquerypatsuffixpercent#1#2{#1\%#2}

\texosquerypatprefixpermill The first argument is the value, the second argument is optional text before the per-mill symbol.

958 \def\texosquerypatprefixpermill#1#2{#2‰#1}

\texosquerypatsuffixpermill The first argument is the value, the second argument is optional text after the per-mill symbol.

959 \def\texosquerypatsuffixpermill#1#2{#1‰#2}

</dl>

`\@texosquery@pattern@shortcuts` Provide much shorter cuts for the convenience of directly defining patterns with `\texosquerydefpattern`.

```
960 \def\@texosquery@pattern@shortcuts{%
961    \def\%{\noexpand\texosquerydtf}%
962    \def\0{\noexpand\texosquerypatdigit}%
963    \def\#{\noexpand\texosquerypatdigitnozero}%
964    \def\-{\noexpand\texosquerypatminus}%
965    \def\,{\noexpand\texosquerypatgroupsep}%
966    \def\numfmt{\noexpand\texosquerypatnum}%
967    \def\pmnumfmt{\noexpand\texosquerypatplusminus}%
968    \def\sinumfmt{\noexpand\texosquerypatsinum}%
969    \def\decfmt{\noexpand\texosquerypatdec}%
970    \def\pcur{\noexpand\texosquerypatprefixcurrency}%
971    \def\picur{\noexpand\texosquerypatprefixicurrency}%
972    \def\scur{\noexpand\texosquerypatsuffixcurrency}%
973    \def\sicur{\noexpand\texosquerypatsuffixicurrency}%
974    \def\ppct{\noexpand\texosquerypatprefixpercent}%
975    \def\spct{\noexpand\texosquerypatsuffixpercent}%
976    \def\ppml{\noexpand\texosquerypatprefixpermill}%
977    \def\spml{\noexpand\texosquerypatsuffixpermill}%
978 }
```

`\texosquerydefpattern` Define a new pattern using the shortcut markup. The first argument is the name of the control sequence in which to store the pattern provided in the second argument. Be careful of any fragile commands within the second argument. They will need protecting!

```
979 \def\texosquerydefpattern#1#2{%
980   \begingroup
981     \@texosquery@pattern@shortcuts
982     \@texosquery@edef\@texosquery@tmp{\endgroup\def\noexpand#1{#2}}\@texosquery@tmp
983 }
```

3.1.4 Applying Date-Time Patterns

In order to apply date-time patterns, we need all the information about the date or time we're trying to format.

1. Era needed by the G designator. Java identifies the era by an integer (0 = BC and 1 = AD).

2. Era text (e.g. AD) can be supplied by a macro.

3. Year needed by the y designator.

4. Week year needed by the Y designator.

5. Month in year needed by the M or L designators.

6. Month name needed by the M or L designators. This can be provided as macros that convert the month number to the name. Four macros are needed: short, full, standalone short and standalone full.

80

7. Week in year needed by the `w` designator.

8. Week in month needed by the `W` designator.

9. Day in year needed by the `D` designator.

10. Day in month needed by the `d` designator.

11. Day of week in month needed by the `F` designator.

12. Day name in week needed by the `E` designator. This can be provided as a macros that accepts the day of week number. The full form is needed for 4 letter patterns otherwise a short form.

13. Day number of week (1 = Monday, 7 = Sunday) needed by the `u` designator. This means that the above day of week name macros needs to use Monday=1 base indexing. This means that `\pgfcalendarweekdayname` can't be used directly.

14. AM/PM identifier needed by the `a` designator. Assume 0 = AM and 1 = PM to match Java.

15. AM/PM text can be provided by a macro.

16. Hour of the day (0-23) needed by the `H` designator.

17. Hour in day (1-24) needed by the `k` designator.

18. Hour in am/pm (0-11) needed by the `K` designator.

19. Hour in am/pm (1-12) needed by the `h` designator.

20. Minute in hour needed by the `m` designator.

21. Second in minute needed by the `s` designator.

22. Millisecond needed by the `S` designator.

23. Time zone needed by the `z`, `Z` and `X` designators. This will require macros for converting the time zone to each of those formats.

Supply a general utility command that has enough arguments to pass all the above information. A higher level user command can then be provided that determines all the arguments to provide an easier interface.

The arguments need to be the pattern followed by {⟨*era id*⟩}{⟨*year*⟩}{⟨*week year*⟩}{⟨*month*⟩}{⟨*week in year*⟩}{⟨*week in month*⟩}{⟨*day in year*⟩}{⟨*day in month*⟩}{⟨*day of week in month*⟩}{⟨*day number of week*⟩}{⟨*am/pm id*⟩}{⟨*hour of day (H)*⟩}{⟨*hour in day (k)*⟩}{⟨*hour in am/pm (K)*⟩}{⟨*hour in am/pm (h)*⟩}{⟨*minute in hour*⟩}{⟨*second in minute*⟩}{⟨*millisecond*⟩}{⟨*time zone*⟩} The arguments must all be integers except for the time zone which must be in the form {⟨*TZh*⟩}{⟨*TZm*⟩}{⟨*id*⟩}{⟨*dst flag*⟩}. These are the time zone hour and min offsets, time zone ID and daylight saving flag (1 if daylight saving in effect otherwise 0). These arguments can all be obtained using the `--date-time` / `-M` action.

We'll need some helper macros to get around the nine argument maximum limit.

\texosqueryfmtdatetime The general utility command to format a pattern. The first argument is the pattern. After that are the date-time data arguments.

```
984 \def\texosqueryfmtdatetime#1{%
985   \def\@texosquery@fmt@dt@pattern{#1}%
986   \@texosquery@fmt@getera
987 }
```

All the remaining arguments except for the time zone must be integers. These are padded using \@texosquery@paddigits.

\@texosquery@fmt@getera

```
988 \def\@texosquery@fmt@getera#1{%
989   \edef\@texosquery@fmt@G{\@texosquery@paddigits{#1}}%
990   \@texosquery@fmt@getyear
991 }
```

\@texosquery@fmt@getyear

```
992 \def\@texosquery@fmt@getyear#1{%
993   \edef\@texosquery@fmt@y{\@texosquery@paddigits{#1}}%
994   \@texosquery@fmt@getweekyear
995 }
```

\@texosquery@fmt@getweekyear

```
996 \def\@texosquery@fmt@getweekyear#1{%
997   \edef\@texosquery@fmt@Y{\@texosquery@paddigits{#1}}%
998   \@texosquery@fmt@getmonth
999 }
```

\@texosquery@fmt@getmonth

```
1000 \def\@texosquery@fmt@getmonth#1{%
1001   \edef\@texosquery@fmt@M{\@texosquery@paddigits{#1}}%
1002   \let\@texosquery@fmt@L\@texosquery@fmt@M
1003   \@texosquery@fmt@getweekinyear
1004 }
```

\@texosquery@fmt@getweekinyear

```
1005 \def\@texosquery@fmt@getweekinyear#1{%
1006   \edef\@texosquery@fmt@w{\@texosquery@paddigits{#1}}%
1007   \@texosquery@fmt@getweekinmonth
1008 }
```

\@texosquery@fmt@getweekinmonth

```
1009 \def\@texosquery@fmt@getweekinmonth#1{%
1010   \edef\@texosquery@fmt@W{\@texosquery@paddigits{#1}}%
1011   \@texosquery@fmt@getdayinyear
1012 }
```

```
1013 \def\@texosquery@fmt@getdayinyear#1{%
1014   \edef\@texosquery@fmt@D{\@texosquery@paddigits{#1}}%
1015   \@texosquery@fmt@getdayinmonth
1016 }
```

```
1017 \def\@texosquery@fmt@getdayinmonth#1{%
1018   \edef\@texosquery@fmt@d{\@texosquery@paddigits{#1}}%
1019   \@texosquery@fmt@getdayofweekinmonth
1020 }
```

```
1021 \def\@texosquery@fmt@getdayofweekinmonth#1{%
1022   \edef\@texosquery@fmt@F{\@texosquery@paddigits{#1}}%
1023   \@texosquery@fmt@getdaynumberofweek
1024 }
```

```
1025 \def\@texosquery@fmt@getdaynumberofweek#1{%
1026   \edef\@texosquery@fmt@u{\@texosquery@paddigits{#1}}%
1027   \let\@texosquery@fmt@E\@texosquery@fmt@u
1028   \@texosquery@fmt@getampm
1029 }
```

```
1030 \def\@texosquery@fmt@getampm#1{%
1031   \edef\@texosquery@fmt@a{\@texosquery@paddigits{#1}}%
1032   \@texosquery@fmt@gethourindayH
1033 }
```

```
1034 \def\@texosquery@fmt@gethourindayH#1{%
1035   \edef\@texosquery@fmt@H{\@texosquery@paddigits{#1}}%
1036   \@texosquery@fmt@gethourindayk
1037 }
```

```
1038 \def\@texosquery@fmt@gethourindayk#1{%
1039   \edef\@texosquery@fmt@k{\@texosquery@paddigits{#1}}%
1040   \@texosquery@fmt@gethourinampmK
1041 }
```

```
1042 \def\@texosquery@fmt@gethourinampmK#1{%
1043   \edef\@texosquery@fmt@K{\@texosquery@paddigits{#1}}%
1044   \@texosquery@fmt@gethourinampmh
1045 }
```

```
1046 \def\@texosquery@fmt@gethourinampmh#1{%
1047   \edef\@texosquery@fmt@h{\@texosquery@paddigits{#1}}%
1048   \@texosquery@fmt@getminute
1049 }
```

```
1050 \def\@texosquery@fmt@getminute#1{%
1051   \edef\@texosquery@fmt@m{\@texosquery@paddigits{#1}}%
1052   \@texosquery@fmt@getsecond
1053 }
```

```
1054 \def\@texosquery@fmt@getsecond#1{%
1055   \edef\@texosquery@fmt@s{\@texosquery@paddigits{#1}}%
1056   \@texosquery@fmt@getmillisecond
1057 }
```

```
1058 \def\@texosquery@fmt@getmillisecond#1{%
1059   \edef\@texosquery@fmt@S{\@texosquery@paddigits{#1}}%
1060   \@texosquery@fmt@gettimezone
1061 }
```

```
1062 \def\@texosquery@fmt@gettimezone#1{%
1063   \def\@texosquery@fmt@Z{#1}%
1064   \def\@texosquery@fmt@z{#1}%
1065   \def\@texosquery@fmt@X{#1}%
```

All data now supplied. Temporarily redefine pattern markup and process the pattern.

```
1066   \begingroup
1067     \@texosquery@setup@dtpattern
1068     \@texosquery@fmt@dt@pattern
1069   \endgroup
1070 }
```

```
1071 \def\@texosquery@setup@dtpattern{%
1072   \let\texosquerydtf\@texosquery@fmt@dtf
1073   \let\texosquerypatstr\texosquerypatfmtstr
1074   \let\texosquerypatquote\texosquerypatfmtquote
1075 }
```

Pad positive number to 10 digits. TEX can't reach 11 digits, so this is the maximum representation.

```
1076 \def\@texosquery@paddigits@pos#1{%
1077   \ifnum#1<10
1078     000000000\number#1
```

```
1079   \else
1080    \ifnum#1<100
1081     00000000\number#1
1082    \else
1083      \ifnum#1<1000
1084       0000000\number#1
1085      \else
1086        \ifnum#1<10000
1087         000000\number#1
1088        \else
1089          \ifnum#1<100000
1090           00000\number#1
1091          \else
1092            \ifnum#1<1000000
1093             0000\number#1
1094            \else
1095              \ifnum#1<10000000
1096               000\number#1
1097              \else
1098                \ifnum#1<100000000
1099                 00\number#1
1100                \else
1101                  \ifnum#1<1000000000
1102                   0\number#1
1103                  \else
1104                    \number#1
1105                  \fi
1106                \fi
1107              \fi
1108            \fi
1109          \fi
1110        \fi
1111      \fi
1112    \fi
1113   \fi
1114 }
```

\@texosquery@paddigits   This will expand to 11 characters (sign followed by 10 digits).

```
1115 \def\@texosquery@paddigits#1{%
1116 \ifnum#1<0
```

Move the minus sign outside.

```
1117   -\expandafter\@texosquery@paddigits@pos\expandafter
1118     {\@texosquery@gobble#1}%
1119 \else
1120   +\@texosquery@paddigits@pos{#1}%
1121 \fi
1122 }
```

\@texosquery@paddigits@trailing   Pad trailing zeros.

```
1123 \def\@texosquery@paddigits@trailing#1{%
1124   \expandafter\@texosquery@tenoften@then@gobble
1125     #10000000000\@texosquery@end@tenoften
1126 }
```

```
1127 \def\@texosquery@tenoften@then@gobble#1#2#3#4#5#6#7#8#9{%
1128   #1#2#3#4#5#6#7#8#9%
1129   \@texosquery@lastoften@gobble
1130 }
```

```
1131 \def\@texosquery@lastoften@gobble#1#2\@texosquery@end@tenoften{#1}
```

Provide commands to select certain digits. (Sign not included.)

First of ten.

```
1132 \def\@texosquery@firstoften#1#2#3#4#5#6#7#8#9{%
1133 #1%
```

Grab tenth argument and discard.

```
1134 \@texosquery@gobble
1135 }
```

Second of ten.

```
1136 \def\@texosquery@secondoften#1#2#3#4#5#6#7#8#9{%
1137 #2%
```

Grab tenth argument and discard.

```
1138 \@texosquery@gobble
1139 }
```

Third of ten.

```
1140 \def\@texosquery@thirdoften#1#2#3#4#5#6#7#8#9{%
1141 #3%
```

Grab tenth argument and discard.

```
1142 \@texosquery@gobble
1143 }
```

Fourth of ten.

```
1144 \def\@texosquery@fourthoften#1#2#3#4#5#6#7#8#9{%
1145 #4%
```

Grab tenth argument and discard.

```
1146 \@texosquery@gobble
1147 }
```

Fifth of ten.

```
1148 \def\@texosquery@fifthoften#1#2#3#4#5#6#7#8#9{%
1149 #5%
```

Grab tenth argument and discard.

```
1150  \@texosquery@gobble
1151 }
```

\@texosquery@sixthoften   Sixth of ten.

```
1152 \def\@texosquery@sixthoften#1#2#3#4#5#6#7#8#9{%
1153  #6%
```

Grab tenth argument and discard.

```
1154  \@texosquery@gobble
1155 }
```

\@texosquery@seventhoften   Seventh of ten.

```
1156 \def\@texosquery@seventhoften#1#2#3#4#5#6#7#8#9{%
1157  #7%
```

Grab tenth argument and discard.

```
1158  \@texosquery@gobble
1159 }
```

\@texosquery@eighthoften   Eighth of ten.

```
1160 \def\@texosquery@eighthoften#1#2#3#4#5#6#7#8#9{%
1161  #8%
```

Grab tenth argument and discard.

```
1162  \@texosquery@gobble
1163 }
```

\@texosquery@ninthoften   Ninth of ten.

```
1164 \def\@texosquery@ninthoften#1#2#3#4#5#6#7#8#9{%
1165  #9%
```

Grab tenth argument and discard.

```
1166  \@texosquery@gobble
1167 }
```

\@texosquery@tenthoften   Tenth of ten.

```
1168 \def\@texosquery@tenthoften#1#2#3#4#5#6#7#8#9{%
1169  \@texosquery@firstofone
1170 }
```

Now macros to select first n of ten.

@texosquery@firsttwooften   First two of ten.

```
1171 \def\@texosquery@firsttwooften#1#2#3#4#5#6#7#8#9{%
1172  #1#2%
```

Grab tenth argument and discard.

```
1173  \@texosquery@gobble
1174 }
```

@texosquery@firstthreeoften First three of ten.

```
1175 \def\@texosquery@firstthreeoften#1#2#3#4#5#6#7#8#9{%
1176 #1#2#3%
```

Grab tenth argument and discard.

```
1177 \@texosquery@gobble
1178 }
```

@texosquery@firstfouroften First four of ten.

```
1179 \def\@texosquery@firstfouroften#1#2#3#4#5#6#7#8#9{%
1180 #1#2#3#4%
```

Grab tenth argument and discard.

```
1181 \@texosquery@gobble
1182 }
```

@texosquery@firstfiveoften First five of ten.

```
1183 \def\@texosquery@firstfiveoften#1#2#3#4#5#6#7#8#9{%
1184 #1#2#3#4#5%
```

Grab tenth argument and discard.

```
1185 \@texosquery@gobble
1186 }
```

@texosquery@firstsixoften First six of ten.

```
1187 \def\@texosquery@firstsixoften#1#2#3#4#5#6#7#8#9{%
1188 #1#2#3#4#5#6%
```

Grab tenth argument and discard.

```
1189 \@texosquery@gobble
1190 }
```

@texosquery@firstsevenoften First seven of ten.

```
1191 \def\@texosquery@firstsevenoften#1#2#3#4#5#6#7#8#9{%
1192 #1#2#3#4#5#6#7%
```

Grab tenth argument and discard.

```
1193 \@texosquery@gobble
1194 }
```

@texosquery@firsteightoften First eight of ten.

```
1195 \def\@texosquery@firsteightoften#1#2#3#4#5#6#7#8#9{%
1196 #1#2#3#4#5#6#7#8%
```

Grab tenth argument and discard.

```
1197 \@texosquery@gobble
1198 }
```

@texosquery@firstnineoften First nine of ten.

```
1199 \def\@texosquery@firstnineoften#1#2#3#4#5#6#7#8#9{%
1200 #1#2#3#4#5#6#7#8#9%
```

Grab tenth argument and discard.

```
1201  \@texosquery@gobble
1202 }
```

@texosquery@alltenoften   All ten.

```
1203 \def\@texosquery@alltenoften#1#2#3#4#5#6#7#8#9{%
1204   #1#2#3#4#5#6#7#8#9%
1205   \@texosquery@firstofone
1206 }
```

Select last n of ten.

@texosquery@lasttwooften   Last two of ten.

```
1207 \def\@texosquery@lasttwooften#1#2#3#4#5#6#7#8#9{%
1208   #9%
1209   \@texosquery@firstofone
1210 }
```

@texosquery@lastthreeoften   Last three of ten.

```
1211 \def\@texosquery@lastthreeoften#1#2#3#4#5#6#7#8#9{%
1212   #8#9%
1213   \@texosquery@firstofone
1214 }
```

@texosquery@lastfouroften   Last four of ten.

```
1215 \def\@texosquery@lastfouroften#1#2#3#4#5#6#7#8#9{%
1216   #7#8#9%
1217   \@texosquery@firstofone
1218 }
```

@texosquery@lastfiveoften   Last five of ten.

```
1219 \def\@texosquery@lastfiveoften#1#2#3#4#5#6#7#8#9{%
1220   #6#7#8#9%
1221   \@texosquery@firstofone
1222 }
```

@texosquery@lastsixoften   Last six of ten.

```
1223 \def\@texosquery@lastsixoften#1#2#3#4#5#6#7#8#9{%
1224   #5#6#7#8#9%
1225   \@texosquery@firstofone
1226 }
```

@texosquery@lastsevenoften   Last seven of ten.

```
1227 \def\@texosquery@lastsevenoften#1#2#3#4#5#6#7#8#9{%
1228   #4#5#6#7#8#9%
1229   \@texosquery@firstofone
1230 }
```

@texosquery@lasteightoften   Last eight of ten.

```
1231 \def\@texosquery@lasteightoften#1#2#3#4#5#6#7#8#9{%
1232  #3#4#5#6#7#8#9%
1233  \@texosquery@firstofone
1234 }
```

@texosquery@lastnineoften   Last nine of ten.

```
1235 \def\@texosquery@lastnineoften#1#2#3#4#5#6#7#8#9{%
1236  #2#3#4#5#6#7#8#9%
1237  \@texosquery@firstofone
1238 }
```

\@texosquery@fmtminus   Minus symbol for use in date-time patterns.

```
1239 \def\@texosquery@fmtminus{\texosquerypatfmtminus}
```

\@texosquery@fmtplus   Plus symbol for use in date-time patterns. Omit by default.

```
1240 \def\@texosquery@fmtplus{}
```

\@texosquery@fmtsign   Plus or minus sign for use in date-time patterns.

```
1241 \def\@texosquery@fmtsign#1{%
1242  \ifx#1+\@texosquery@fmtplus\else\@texosquery@fmtminus\fi
1243 }
```

\@texosquery@atleastonedigit   At least one digit with leading zeros removed.

```
1244 \def\@texosquery@atleastonedigit#1{%
1245   \ifnum#1<0
1246     \@texosquery@fmtminus\number-#1
1247   \else
1248     \number#1
1249   \fi
1250 }
```

\@texosquery@atleastfourdigits   At least four digits, possible padded with zeros to make up four. The first argument is the
                                   sign, then follow the ten digits.

```
1251 \def\@texosquery@atleastfourdigits#1{%
1252   \@texosquery@at@leastfourdigits#1\@texosquery@end@atleastfourdigits
1253 }
```

\@texosquery@at@leastfourdigits   At least four digits, possible padded with zeros to make up four. The first argument is the
                                   sign, then follow the ten digits.

```
1254 \def\@texosquery@at@leastfourdigits#1#2\@texosquery@end@atleastfourdigits{%
1255   \@texosquery@fmtsign{#1}%
1256   \ifnum#2<1000
1257     \@texosquery@lastfouroften#2%
1258   \else
1259     \number#2
1260   \fi
1261 }
```

\@texosquery@threedigitsexactly Exactly three digits.

```
1262 \def\@texosquery@threedigitsexactly#1{%
1263   \@texosquery@threedigits@exactly#1\@texosquery@threedigits@exactly
1264 }%
```

@texosquery@threedigits@exactly Exactly three digits.

```
1265 \def\@texosquery@threedigits@exactly#1#2\@texosquery@threedigits@exactly{%
1266   \@texosquery@fmtsign{#1}%
1267   \@texosquery@lastthreeoften#2%
1268 }%
```

\@texosquery@twodigitsexactly Exactly two digits.

```
1269 \def\@texosquery@twodigitsexactly#1{%
1270   \@texosquery@twodigits@exactly#1\@texosquery@twodigits@exactly
1271 }%
```

\@texosquery@twodigits@exactly Exactly two digits.

```
1272 \def\@texosquery@twodigits@exactly#1#2\@texosquery@twodigits@exactly{%
1273   \@texosquery@fmtsign{#1}%
1274   \@texosquery@lasttwooften#2%
1275 }%
```

\@texosquery@fmt@dtf \@texosquery@fmt@dtf{⟨*n*⟩}{⟨*designator*⟩}

When formatting a date-time pattern \texosquerydtf will temporarily be rede-
fined to this command. This command indicates the format obtained by ⟨*n*⟩ instances
of ⟨*designator*⟩. For example, {2}{M} indicates the format MM. This command tests for
\texosqueryfmtpat⟨*format*⟩, which should take a single argument. If defined, that's
used, otherwise use one of the numeric commands defined above. The tex-locale package
defines \texosqueryfmtpatMMM and so on to use the locale's month names etc.

```
1276 \def\@texosquery@fmt@dtf#1#2{%
1277   \@texosquery@ifundef{@texosquery@fmt@#2}%
1278   {\@texosquery@warn{Unknown date-time pattern designator '#2'}}%
1279   {%
1280     \ifcase#1
1281     \or
1282       \@texosquery@ifundef{texosqueryfmtpat#2}%
1283       {%
1284         \expandafter\expandafter\expandafter
1285          \@texosquery@atleastonedigit
1286          \expandafter\expandafter\expandafter
1287         {\csname @texosquery@fmt@#2\endcsname}%
1288       }%
1289       {%
1290         \csname texosqueryfmtpat#2\expandafter\expandafter\expandafter\endcsname
1291          \expandafter\expandafter\expandafter
1292         {\csname @texosquery@fmt@#2\endcsname}%
1293       }%
1294     \or
```

```
1295        \@texosquery@ifundef{texosqueryfmtpat#2#2}%
1296        {%
1297          \expandafter\expandafter\expandafter
1298            \@texosquery@twodigitsexactly
1299            \expandafter\expandafter\expandafter
1300            {\csname @texosquery@fmt@#2\endcsname}%
1301        }%
1302        {%
1303          \csname texosqueryfmtpat#2#2\expandafter\expandafter\expandafter
1304          \endcsname \expandafter\expandafter\expandafter
1305            {\csname @texosquery@fmt@#2\endcsname}%
1306        }%
1307      \or
1308        \@texosquery@ifundef{texosqueryfmtpat#2#2#2}%
1309        {%
1310          \expandafter\expandafter\expandafter
1311            \@texosquery@threedigitsexactly
1312            \expandafter\expandafter\expandafter
1313          {\csname @texosquery@fmt@#2\endcsname}%
1314        }%
1315        {%
1316          \csname texosqueryfmtpat#2#2#2\expandafter\expandafter\expandafter
1317          \endcsname \expandafter\expandafter\expandafter
1318            {\csname @texosquery@fmt@#2\endcsname}%
1319        }%
1320      \else
1321        \@texosquery@ifundef{texosqueryfmtpat#2#2#2#2}%
1322        {%
1323          \expandafter\expandafter\expandafter
1324            \@texosquery@atleastfourdigits
1325            \expandafter\expandafter\expandafter
1326            {\csname @texosquery@fmt@#2\endcsname}%
1327        }%
1328        {%
1329          \csname texosqueryfmtpat#2#2#2#2\expandafter\expandafter\expandafter
1330          \endcsname \expandafter\expandafter\expandafter
1331            {\csname @texosquery@fmt@#2\endcsname}%
1332        }%
1333      \fi
1334    }%
1335 }
```

Provide default commands for the time zone designators, since the time zone isn't supplied as a single integer.

`\texosqueryfmttimezonehr` Allow for -0 so append 1 to hour in test.

```
1336 \def\texosqueryfmttimezonehr#1{%
1337   \ifnum#11<0\@texosquery@fmtminus
1338     \ifnum#1>-10 0\fi\number-#1
1339   \else
```

```
1340      +\ifnum#1<10 0\fi\number#1
1341   \fi
1342 }
```

\texosqueryfmttimezonenumhr   Like the above, but don't zero-pad or prefix with plus sign.

```
1343 \def\texosqueryfmttimezonenumhr#1{%
1344   \ifnum#11<0\@texosquery@fmtminus
1345     \number-#1
1346   \else
1347     \number#1
1348   \fi
1349 }
```

\texosqueryfmttimezonemin

```
1350 \def\texosqueryfmttimezonemin#1{%
1351   \ifnum#1<10 0\fi\number#1
1352 }
```

\@texosquery@firstoffour

```
1353 \def\@texosquery@firstoffour#1#2#3#4{#1}
```

\@texosquery@secondoffour

```
1354 \def\@texosquery@secondoffour#1#2#3#4{#2}
```

\@texosquery@thirdoffour

```
1355 \def\@texosquery@thirdoffour#1#2#3#4{#3}
```

\@texosquery@fourthoffour

```
1356 \def\@texosquery@fourthoffour#1#2#3#4{#4}
```

\texosqueryshorttimezone   Maps id to short time zone display name. This will need redefining as appropriate. The default simply expands to the ID. Mappings can be obtained for a particular locale using the -Z or --time-zones action.

```
1357 \def\texosqueryshorttimezone#1{#1}
```

\texosqueryshortdstzone   Maps id to short daylight saving time zone display name. This will need redefining as appropriate. The default simply expands to the ID followed by (DST).

```
1358 \def\texosqueryshortdstzone#1{#1 (DST)}
```

\texosquerylongtimezone   Maps id to long time zone display name. This will need redefining as appropriate. The default simply expands to the ID.

```
1359 \def\texosquerylongtimezone#1{#1}
```

\texosquerylongdstzone   Maps id to long daylight saving time zone display name. This will need redefining as appropriate. The default simply expands to the ID followed by (DST).

```
1360 \def\texosquerylongdstzone#1{#1 (DST)}
```

`\texosquerytimesep`

```
1361 \def\texosquerytimesep{:}
```

`\texosqueryfmtpatz` Default time zone format for z designator.

```
1362 \def\texosqueryfmtpatz#1{%
1363 \expandafter\ifnum\@texosquery@fourthoffour#1=0
1364   \expandafter\texosqueryshorttimezone\expandafter{\@texosquery@thirdoffour#1}%
1365 \else
1366   \expandafter\texosqueryshortdstzone\expandafter{\@texosquery@thirdoffour#1}%
1367 \fi
1368 }
```

`\texosqueryfmtpatzz` Default time zone format for zz designator.

```
1369 \def\texosqueryfmtpatzz#1{%
1370 \expandafter\ifnum\@texosquery@fourthoffour#1=0
1371   \expandafter\texosqueryshorttimezone\expandafter{\@texosquery@thirdoffour#1}%
1372 \else
1373   \expandafter\texosqueryshortdstzone\expandafter{\@texosquery@thirdoffour#1}%
1374 \fi
1375 }
```

`\texosqueryfmtpatzzz` Default time zone format for zzz designator.

```
1376 \def\texosqueryfmtpatzzz#1{%
1377 \expandafter\ifnum\@texosquery@fourthoffour#1=0
1378   \expandafter\texosquerylongtimezone\expandafter{\@texosquery@thirdoffour#1}%
1379 \else
1380   \expandafter\texosquerylongdstzone\expandafter{\@texosquery@thirdoffour#1}%
1381 \fi
1382 }
```

`\texosqueryfmtpatzzzz` Default time zone format for zzzz designator.

```
1383 \def\texosqueryfmtpatzzzz#1{%
1384 \expandafter\ifnum\@texosquery@fourthoffour#1=0
1385   \expandafter\texosquerylongtimezone\expandafter{\@texosquery@thirdoffour#1}%
1386 \else
1387   \expandafter\texosquerylongdstzone\expandafter{\@texosquery@thirdoffour#1}%
1388 \fi
1389 }
```

`\texosqueryfmtpatZ` Default time zone format for Z designator.

```
1390 \def\texosqueryfmtpatZ#1{%
1391   \expandafter\texosqueryfmttimezonehr\expandafter
1392   {\@texosquery@firstoffour#1}%
1393   \expandafter\texosqueryfmttimezonemin\expandafter
1394   {\@texosquery@secondoffour#1}%
1395 }
```

`\texosqueryfmtpatZZ` Default time zone format for ZZ designator.

```
1396 \def\texosqueryfmtpatZZ#1{%
```

```
1397    \expandafter\texosqueryfmttimezonehr\expandafter
1398      {\@texosquery@firstoffour#1}%
1399    \expandafter\texosqueryfmttimezonemin\expandafter
1400      {\@texosquery@secondoffour#1}%
1401 }
```

\texosqueryfmtpatZZZ   Default time zone format for ZZZ designator.

```
1402 \def\texosqueryfmtpatZZZ#1{%
1403    \expandafter\texosqueryfmttimezonehr\expandafter
1404      {\@texosquery@firstoffour#1}%
1405    \expandafter\texosqueryfmttimezonemin\expandafter
1406      {\@texosquery@secondoffour#1}%
1407 }
```

\texosqueryfmtpatZZZZ   Default time zone format for ZZZZ designator.

```
1408 \def\texosqueryfmtpatZZZZ#1{%
1409    \expandafter\texosqueryfmttimezonehr\expandafter
1410      {\@texosquery@firstoffour#1}%
1411    \expandafter\texosqueryfmttimezonemin\expandafter
1412      {\@texosquery@secondoffour#1}%
1413 }
```

\texosqueryfmtpatX   Default time zone format for X designator.

```
1414 \def\texosqueryfmtpatX#1{%
1415    \expandafter\texosqueryfmttimezonehr\expandafter
1416      {\@texosquery@firstoffour#1}%
1417    \texosquerytimesep
1418    \expandafter\texosqueryfmttimezonemin\expandafter
1419      {\@texosquery@secondoffour#1}%
1420 }
```

\texosqueryfmtpatXX   Default time zone format for XX designator.

```
1421 \def\texosqueryfmtpatXX#1{%
1422    \expandafter\texosqueryfmttimezonehr\expandafter
1423      {\@texosquery@firstoffour#1}%
1424    \texosquerytimesep
1425    \expandafter\texosqueryfmttimezonemin\expandafter
1426      {\@texosquery@secondoffour#1}%
1427 }
```

\texosqueryfmtpatXXX   Default time zone format for XXX designator.

```
1428 \def\texosqueryfmtpatXXX#1{%
1429    \expandafter\texosqueryfmttimezonehr\expandafter
1430      {\@texosquery@firstoffour#1}%
1431    \texosquerytimesep
1432    \expandafter\texosqueryfmttimezonemin\expandafter
1433      {\@texosquery@secondoffour#1}%
1434 }
```

`\texosqueryfmtpatXXXX`   Default time zone format for XXXX designator.

```
1435 \def\texosqueryfmtpatXXXX#1{%
1436   \expandafter\texosqueryfmttimezonehr\expandafter
1437     {\@texosquery@firstoffour#1}%
1438   \texosquerytimesep
1439   \expandafter\texosqueryfmttimezonemin\expandafter
1440     {\@texosquery@secondoffour#1}%
1441 }
```

`\texosqueryfmtpata`   Default am/pm designator for the a designator.

```
1442 \def\texosqueryfmtpata#1{%
1443   \ifnum#1=0 AM\else PM\fi
1444 }
```

`\texosqueryfmtpataa`   Default am/pm for the aa designator. Just make it the same as the a designator.

```
1445 \def\texosqueryfmtpataa{\texosqueryfmtpata}
```

`\texosqueryfmtpataaa`   Default am/pm for the aaa designator. Just make it the same as the a designator.

```
1446 \def\texosqueryfmtpataaa{\texosqueryfmtpata}
```

`\texosqueryfmtpataaaa`   Default am/pm for the aaaa designator. Just make it the same as the a designator.

```
1447 \def\texosqueryfmtpataaaa{\texosqueryfmtpata}
```

`\texosqueryfmtpatG`   Default era designator for the G designator.

```
1448 \def\texosqueryfmtpatG#1{%
1449   \ifnum#1=1 AD\else BC\fi
1450 }
```

`\texosqueryfmtpatGG`   Default era for the GG designator. Just make it the same as the G designator.

```
1451 \def\texosqueryfmtpatGG{\texosqueryfmtpatG}
```

`\texosqueryfmtpatGGG`   Default era for the GGG designator. Just make it the same as the G designator.

```
1452 \def\texosqueryfmtpatGGG{\texosqueryfmtpatG}
```

`\texosqueryfmtpatGGGG`   Default era for the GGGG designator. Just make it the same as the G designator.

```
1453 \def\texosqueryfmtpatGGGG{\texosqueryfmtpatG}
```

3.1.5 Applying Numeric Patterns

`\texosqueryfmtnumber`   `\texosqueryfmtnumber{`⟨*pattern*⟩`}{`⟨*int*⟩`}{`⟨*frac*⟩`}{`⟨*mantissa*⟩`}`

General purpose low-level number formatting command. The first argument ⟨*pattern*⟩ is the pattern. The other arguments are unformatted integers and must be present and not exceed 10 digits each. The ⟨*frac*⟩ part must not start with a sign. The minus sign should go at the start of ⟨*int*⟩ for negative numbers. The plus sign is optional for positive ⟨*int*⟩ or ⟨*mantissa*⟩ and not permitted in ⟨*frac*⟩. The arguments may each be the actual numerical value or be a single control sequence whose replacement text is the value. Avoid anything more complicated than that.

This package doesn't provide a higher level command that can split a number into integer, fractional and mantissa parts.

```
1454 \def\texosqueryfmtnumber#1#2#3#4{%
1455   \begingroup
1456   \let\texosquerypatstr\texosquerypatfmtstr
1457   \let\texosquerypatquote\texosquerypatfmtquote
1458   \let\texosquerypatplusminus\texosquerypatfmt@plusminus
1459   \let\texosquerypatnum\texosquerypatfmt@num
1460   \let\texosquerypatsinum\texosquerypatfmt@sinum
1461   \let\texosquerypatdec\texosquerypatfmt@dec
1462   \let\texosquerypatprefixcurrency\texosquery@patfmt@prefixcurrency
1463   \let\texosquerypatprefixicurrency\texosquery@patfmt@prefixicurrency
1464   \let\texosquerypatsuffixcurrency\texosquery@patfmt@suffixcurrency
1465   \let\texosquerypatsuffixicurrency\texosquery@patfmt@suffixicurrency
1466   \let\texosquerypatdigit\texosquerypatfmt@digit
1467   \let\texosquerypatdigitnozero\texosquerypatfmt@digitnozero
1468   \let\texosquerypatgroupsep\texosquerypatfmt@groupsep
1469   \let\texosquerypatprefixpercent\texosquery@patfmt@prefixpercent
1470   \let\texosquerypatsuffixpercent\texosquery@patfmt@suffixpercent
1471   \let\texosquerypatprefixpermill\texosquery@patfmt@prefixpermill
1472   \let\texosquerypatsuffixpermill\texosquery@patfmt@suffixpermill
1473   \let\texosquerypatminus\@texosquerypat@numfmt@sign
1474   \let\texosquerypatfmt@decsep\texosquerypatfmtdecsep
```

Allow for negative zero in the ⟨int⟩ part. To avoid overflow, first check for 0 and then append 1 to the number to catch -0.

```
1475   \edef\@texosquery@sgn{%
1476     \ifnum#2=0
1477       \expandafter\ifnum#21<0 -\else+\fi
1478     \else
1479       \ifnum#2<0 -\else+\fi
1480     \fi
1481   }%
```

Allow for arguments passed as control sequences that expand to a number.

```
1482   \edef\@texosquery@int{\expandafter\@texosquery@paddigits
1483     \expandafter{\number#2}}%
1484   \let\@texosquery@si@int\@texosquery@int
```

Can't use \number here as we'll lose any leading zeros.

```
1485   \edef\@texosquery@frac{\expandafter\@texosquery@paddigits@trailing
1486     \expandafter{#3}}%
1487   \let\@texosquery@si@frac\@texosquery@frac
1488   \edef\@texosquery@mantissa{\expandafter\@texosquery@paddigits
1489     \expandafter{\number#4}}%
```

Is the mantissa non-zero?

```
1490   \ifnum#4=0\relax
1491   \else
1492     \expandafter\ifx\@texosquery@sgn-%
1493       \edef\@texosquery@int{\expandafter
```

```
1494            \@texosquery@paddigits@pos\expandafter{\number-#2}}%
1495          \else
1496            \edef\@texosquery@int{\@texosquery@paddigits@pos{#2}}%
1497          \fi
```
Shift.
```
1498          \ifnum#4<0
1499            \expandafter\@texosquery@neg@shift\expandafter{\number-#4}%
1500          \else
1501            \@texosquery@pos@shift{#4}%
1502          \fi
1503          \expandafter\ifx\@texosquery@sgn-%
1504            \edef\@texosquery@int{\expandafter\@texosquery@paddigits
1505              \expandafter{\number-\@texosquery@int}}%
1506          \else
1507            \edef\@texosquery@int{\expandafter\@texosquery@paddigits
1508              \expandafter{\number\@texosquery@int}}%
1509          \fi
1510          \edef\@texosquery@frac{\@texosquery@paddigits@trailing{\@texosquery@frac}}%
1511        \fi
1512        \edef\@texosquery@current{\expandafter\@texosquery@gobble\@texosquery@int}%
1513        \let\@texosquery@zerodigit\@texosquery@zerodigit@leading
1514        \expandafter\ifx\@texosquery@sgn-%
1515          \let\@texosquery@currentsign\texosquerypatfmtminus
1516        \else
1517          \let\@texosquery@currentsign\texosquerypatfmtplus
1518        \fi
1519        \@texosquery@digitindex=0\relax
1520        \let\@texosquery@patfmt@dosep\empty
1521        \@texosquery@digitfoundfalse
1522        #1%
1523      \endgroup
1524 }
```

\@texosquery@digitindex
```
1525 \newcount\@texosquery@digitindex
```

\if@texosquery@digitfound
```
1526 \newif\if@texosquery@digitfound
```

Macros to shift the decimal place.

\@texosquery@pos@shift
```
1527 \def\@texosquery@pos@shift#1{%
1528    \ifcase#1
1529    \or
1530      \edef\@texosquery@int{%
1531        \expandafter\@texosquery@lastnineoften\@texosquery@int
1532        \expandafter\@texosquery@firstoften\@texosquery@frac}%
1533      \edef\@texosquery@frac{%
```

```
1534        \expandafter\@texosquery@lastnineoften\@texosquery@frac
1535      }%
1536  \or
1537    \edef\@texosquery@int{%
1538      \expandafter\@texosquery@lasteightoften\@texosquery@int
1539      \expandafter\@texosquery@firsttwooften\@texosquery@frac}%
1540    \edef\@texosquery@frac{%
1541        \expandafter\@texosquery@lasteightoften\@texosquery@frac
1542    }%
1543  \or
1544    \edef\@texosquery@int{%
1545      \expandafter\@texosquery@lastsevenoften\@texosquery@int
1546      \expandafter\@texosquery@firstthreeoften\@texosquery@frac}%
1547    \edef\@texosquery@frac{%
1548        \expandafter\@texosquery@lastsevenoften\@texosquery@frac
1549    }%
1550  \or
1551    \edef\@texosquery@int{%
1552      \expandafter\@texosquery@lastsixoften\@texosquery@int
1553      \expandafter\@texosquery@firstfouroften\@texosquery@frac}%
1554    \edef\@texosquery@frac{%
1555        \expandafter\@texosquery@lastsixoften\@texosquery@frac
1556    }%
1557  \or
1558    \edef\@texosquery@int{%
1559      \expandafter\@texosquery@lastfiveoften\@texosquery@int
1560      \expandafter\@texosquery@firstfiveoften\@texosquery@frac}%
1561    \edef\@texosquery@frac{%
1562        \expandafter\@texosquery@lastfiveoften\@texosquery@frac
1563    }%
1564  \or
1565    \edef\@texosquery@int{%
1566      \expandafter\@texosquery@lastfouroften\@texosquery@int
1567      \expandafter\@texosquery@firstsixoften\@texosquery@frac}%
1568    \edef\@texosquery@frac{%
1569        \expandafter\@texosquery@lastfouroften\@texosquery@frac
1570    }%
1571  \or
1572    \edef\@texosquery@int{%
1573      \expandafter\@texosquery@lastthreeoften\@texosquery@int
1574      \expandafter\@texosquery@firstsevenoften\@texosquery@frac}%
1575    \edef\@texosquery@frac{%
1576        \expandafter\@texosquery@lastthreeoften\@texosquery@frac
1577    }%
1578  \or
1579    \edef\@texosquery@int{%
1580      \expandafter\@texosquery@lasttwooften\@texosquery@int
1581      \expandafter\@texosquery@firsteightoften\@texosquery@frac}%
1582    \edef\@texosquery@frac{%
1583        \expandafter\@texosquery@lasttwooften\@texosquery@frac
```

```
1584      }%
1585   \or
1586     \edef\@texosquery@int{%
1587       \expandafter\@texosquery@tenthoften\@texosquery@int
1588       \expandafter\@texosquery@firstnineoften\@texosquery@frac}%
1589     \edef\@texosquery@frac{%
1590        \expandafter\@texosquery@tenthoften\@texosquery@frac
1591     }%
1592   \or
1593     \let\@texosquery@int\@texosquery@frac
1594     \edef\@texosquery@frac{0}%
```

Anything larger will require scientific notation. Hopefully the pattern supports this.

```
1595   \fi
1596 }
```

\@texosquery@neg@shift

```
1597 \def\@texosquery@neg@shift#1{%
1598   \ifcase#1
1599   \or
1600     \edef\@texosquery@frac{%
1601       \expandafter\@texosquery@lastoneoften\@texosquery@int
1602       \expandafter\@texosquery@firstnineoften\@texosquery@frac
1603     }%
1604     \edef\@texosquery@int{%
1605       \expandafter\@texosquery@firstnineoften\@texosquery@int
1606     }%
1607   \or
1608     \edef\@texosquery@frac{%
1609       \expandafter\@texosquery@lasttwooften\@texosquery@int
1610       \expandafter\@texosquery@firsteightoften\@texosquery@frac
1611     }%
1612     \edef\@texosquery@int{%
1613       \expandafter\@texosquery@firsteightoften\@texosquery@int
1614     }%
1615   \or
1616     \edef\@texosquery@frac{%
1617       \expandafter\@texosquery@lastthreeoften\@texosquery@int
1618       \expandafter\@texosquery@firstsevenoften\@texosquery@frac
1619     }%
1620     \edef\@texosquery@int{%
1621       \expandafter\@texosquery@firstsevenoften\@texosquery@int
1622     }%
1623   \or
1624     \edef\@texosquery@frac{%
1625       \expandafter\@texosquery@lastfouroften\@texosquery@int
1626       \expandafter\@texosquery@firstsixoften\@texosquery@frac
1627     }%
1628     \edef\@texosquery@int{%
1629       \expandafter\@texosquery@firstsixoften\@texosquery@int
```

```
1630      }%
1631   \or
1632     \edef\@texosquery@frac{%
1633       \expandafter\@texosquery@lastfiveoften\@texosquery@int
1634       \expandafter\@texosquery@firstfiveoften\@texosquery@frac
1635     }%
1636     \edef\@texosquery@int{%
1637       \expandafter\@texosquery@firstfiveoften\@texosquery@int
1638     }%
1639   \or
1640     \edef\@texosquery@frac{%
1641       \expandafter\@texosquery@lastsixoften\@texosquery@int
1642       \expandafter\@texosquery@firstfouroften\@texosquery@frac
1643     }%
1644     \edef\@texosquery@int{%
1645       \expandafter\@texosquery@firstfouroften\@texosquery@int
1646     }%
1647   \or
1648     \edef\@texosquery@frac{%
1649       \expandafter\@texosquery@lastsevenoften\@texosquery@int
1650       \expandafter\@texosquery@firstthreeoften\@texosquery@frac
1651     }%
1652     \edef\@texosquery@int{%
1653       \expandafter\@texosquery@firstthreeoften\@texosquery@int
1654     }%
1655   \or
1656     \edef\@texosquery@frac{%
1657       \expandafter\@texosquery@lasteightoften\@texosquery@int
1658       \expandafter\@texosquery@firsttwooften\@texosquery@frac
1659     }%
1660     \edef\@texosquery@int{%
1661       \expandafter\@texosquery@firsttwooften\@texosquery@int
1662     }%
1663   \or
1664     \edef\@texosquery@frac{%
1665       \expandafter\@texosquery@lastnineoften\@texosquery@int
1666       \expandafter\@texosquery@firstoften\@texosquery@frac
1667     }%
1668     \edef\@texosquery@int{%
1669       \expandafter\@texosquery@firstoften\@texosquery@int
1670     }%
1671   \or
1672     \edef\@texosquery@frac{\@texosquery@int\@texosquery@frac}%
1673     \edef\@texosquery@int{0}%
```

Anything beyond this will require scientific notation. Hopefully the pattern supports it.

```
1674   \fi
1675 }
```

\@texosquerypat@numfmt@sign

```
1676 \def\@texosquerypat@numfmt@sign{%
1677   \@texosquery@currentsign
1678   \let\@texosquery@currentsign\empty
1679 }
```

\texosquerypatfmtstr

```
1680 \def\texosquerypatfmtstr#1{#1}
```

\texosquerypatfmtquote

```
1681 \def\texosquerypatfmtquote{'}
```

\texosquerypatfmt@plusminus

```
1682 \def\texosquerypatfmt@plusminus#1#2{%
1683  \edef\@texosquery@current{\expandafter\@texosquery@gobble\@texosquery@int}%
1684  \@texosquery@digitindex=0\relax
1685  \let\@texosquery@patfmt@dosep\empty
1686  \@texosquery@digitfoundfalse
1687  \expandafter\ifx\@texosquery@sgn-%
1688   #2%
1689   \ifnum\@texosquery@digitindex=10
1690   \else
1691    \@texosquery@invalidpattern{#2}%
1692   \fi
1693  \else
1694   #1%
1695   \ifnum\@texosquery@digitindex=10
1696   \else
1697    \@texosquery@invalidpattern{#1}%
1698   \fi
1699  \fi
1700 }
```

\texosquerypatfmt@num

```
1701 \def\texosquerypatfmt@num#1{#1}
```

\texosquerypatfmtexp   Exponent symbol. Change as appropriate.

```
1702 \def\texosquerypatfmtexp{E}
```

\texosquerypatfmt@sinum

```
1703 \def\texosquerypatfmt@sinum#1#2{%
1704  \let\@texosquery@int\@texosquery@si@int
1705  \let\@texosquery@frac\@texosquery@si@frac
1706  \let\@texosquery@current\@texosquery@int
1707  #1%
1708  \texosquerypatfmtexp
1709  {\let\@texosquery@zerodigit\@texosquery@zerodigit@leading
1710  \ifnum\@texosquery@mantissa<0\relax
1711    \let\@texosquery@currentsign\texosquerypatfmtminus
1712  \else
```

```
1713     \let\@texosquery@currentsign\texosquerypatfmtplus
1714  \fi
1715  \edef\@texosquery@current{\expandafter\@texosquery@gobble\@texosquery@mantissa}%
1716  \@texosquery@digitindex=0\relax
1717  \let\@texosquery@patfmt@dosep\empty
1718  \@texosquery@digitfoundfalse
1719  #2}}
```

\texosquerypatfmtdecsep   Decimal separator. Change as appropriate.

```
1720 \def\texosquerypatfmtdecsep{.}
```

\texosquerypatfmtcurdecsep   Currency decimal separator. Change as appropriate.

```
1721 \def\texosquerypatfmtcurdecsep{.}
```

\texosquerypatfmt@dec

```
1722 \def\texosquerypatfmt@dec#1#2{%
1723  \edef\@texosquery@current{\expandafter\@texosquery@gobble\@texosquery@int}%
1724  \@texosquery@digitindex=0\relax
1725  \let\@texosquery@patfmt@dosep\empty
1726  \@texosquery@digitfoundfalse
1727  \let\@texosquery@zerodigit\@texosquery@zerodigit@leading
1728  #1%
1729  \ifnum\@texosquery@digitindex=10
1730  \else
1731   \@texosquery@invalidpattern{#1}%
1732  \fi
```

We can't display the decimal separator just yet as the pattern may not show any digits for the fractional part. So only do the separator just before the first digit.

```
1733  \let\@texosquery@patfmt@dosep\texosquerypatfmt@decsep
1734  \let\@texosquery@current\@texosquery@frac
1735  \@texosquery@digitindex=0\relax
1736  \@texosquery@digitfoundfalse
1737  \let\@texosquery@zerodigit\@texosquery@zerodigit@trailing
1738  \let\@texosquery@currentsign\empty
1739  #2%
1740  \ifnum\@texosquery@digitindex=10
1741  \else
1742   \@texosquery@invalidpattern{#2}%
1743  \fi
1744 }
```

\texosquerypatfmtint

```
1745 \def\texosquerypatfmtint#1{%
1746  \edef\@texosquery@current{\expandafter\@texosquery@gobble\@texosquery@int}%
1747  \@texosquery@digitindex=0\relax
1748  \@texosquery@digitfoundfalse
1749  \let\@texosquery@zerodigit\@texosquery@zerodigit@leading
1750  #1%
1751  \ifnum\@texosquery@digitindex=10
```

```
1752  \else
1753    \@texosquery@invalidpattern{#1}%
1754  \fi
1755 }
```

`\@texosquery@setpatdisplay`

```
1756 \def\@texosquery@setpatdisplay{%
1757    \def\texosquerypatstr##1{'##1'}%
1758    \def\texosquerypatquote{''}%
1759    \def\texosquerypatplusminus##1##2{##1;##2}%
1760    \def\texosquerypatnum##1{##1}%
1761    \def\texosquerypatsinum##1##2{##1E##2}%
1762    \def\texosquerypatdec##1##2{##1.##2}%
1763    \def\texosquerypatprefixcurrency##1##2{##2¤##1}%
1764    \def\texosquerypatprefixicurrency##1##2{##2¤¤##1}%
1765    \def\texosquerypatsuffixcurrency##1##2{##1¤##2}%
1766    \def\texosquerypatsuffixicurrency##1##2{##1¤¤##2}%
1767    \def\texosquerypatdigit{0}%
1768    \def\texosquerypatdigitnozero{\#}%
1769    \def\texosquerypatminus{-}%
1770    \def\texosquerypatgroupsep{,}%
1771    \def\texosquerypatprefixpercent##1##2{##2\%##1}%
1772    \def\texosquerypatsuffixpercent##1##2{##1\%##2}%
1773    \def\texosquerypatprefixpermill##1##2{##2‰##1}%
1774    \def\texosquerypatsuffixpermill##1##2{##1‰##2}%
1775    \def\texosquerypatfmt@decsep{.}%
1776 }
```

`\@texosquery@invalidpattern`

```
1777 \def\@texosquery@invalidpattern#1{%
1778    \begingroup
1779      \@texosquery@setpatdisplay
1780      \@texosquery@err{10 digit specifiers expected in
1781        numeric pattern #1. Found \number\@texosquery@digitindex}%
1782       {Each integer element of a numeric pattern must have exactly
1783         10 digit specifiers (0 or \#)}%
1784    \endgroup
1785 }
```

`\texosquerypatfmtcurrencysign` Currency symbol. Redefine as appropriate.

```
1786 \def\texosquerypatfmtcurrencysign{\$}
```

`\texosquerypatfmticurrencysign` International currency symbol. There's no generic fallback that's independent of the input encoding, so this uses a UTF-8 character on the assumption that if \textcurrency isn't available (for example, through textcomp, then the user may be using X∃TEX or LuaTEX). If this isn't the case, and there's no UTF-8 support, then this command will need to be redefined as appropriate.

```
1787 \ifx\textcurrency\undefined
1788    \def\texosquerypatfmticurrencysign{¤}
```

```
1789 \else
1790   \def\texosquerypatfmticurrencysign{\textcurrency}
1791 \fi
```

```
1792 \def\texosquery@patfmt@prefixcurrency#1#2{%
1793 \let\texosquerypatfmt@decsep\texosquerypatfmtcurdecsep
1794 #2\texosquerypatfmtcurrencysign#1%
1795 }
```

```
1796 \def\texosquery@patfmt@prefixicurrency#1#2{%
1797 \let\texosquerypatfmt@decsep\texosquerypatfmtcurdecsep
1798 #2\texosquerypatfmticurrencysign#1%
1799 }
```

```
1800 \def\texosquery@patfmt@suffixcurrency#1#2{%
1801 \let\texosquerypatfmt@decsep\texosquerypatfmtcurdecsep
1802 #1\texosquerypatfmtcurrencysign#2%
1803 }
```

```
1804 \def\texosquery@patfmt@suffixicurrency#1#2{%
1805 \let\texosquerypatfmt@decsep\texosquerypatfmtcurdecsep
1806 #1\texosquerypatfmticurrencysign#2%
1807 }
```

```
1808 \def\texosquerypatfmt@digit{%
1809 \advance\@texosquery@digitindex by 1\relax
1810 \if@texosquery@digitfound
1811 \else
1812   \ifx\@texosquery@currentsign\texosquerypatfmtminus
1813     \texosquerypatfmtminus
1814     \let\@texosquery@currentsign\empty
1815   \else
1816     \@texosquery@patfmt@dosep
1817     \let\@texosquery@patfmt@dosep\empty
1818   \fi
1819 \fi
1820 \@texosquery@digitfoundtrue
1821 \ifcase\@texosquery@digitindex
1822 \or
1823   \expandafter\@texosquery@firstoften\@texosquery@current
1824 \or
1825   \expandafter\@texosquery@secondoften\@texosquery@current
1826 \or
1827   \expandafter\@texosquery@thirdoften\@texosquery@current
```

```
1828 \or
1829   \expandafter\@texosquery@fourthoften\@texosquery@current
1830 \or
1831   \expandafter\@texosquery@fifthoften\@texosquery@current
1832 \or
1833   \expandafter\@texosquery@sixthoften\@texosquery@current
1834 \or
1835   \expandafter\@texosquery@seventhoften\@texosquery@current
1836 \or
1837   \expandafter\@texosquery@eighthoften\@texosquery@current
1838 \or
1839   \expandafter\@texosquery@ninthoften\@texosquery@current
1840 \or
1841   \expandafter\@texosquery@tenthoften\@texosquery@current
1842 \fi
1843 }
```

\texosquerypatfmt@digitnozero

```
1844 \def\texosquerypatfmt@digitnozero{%
1845 \advance\@texosquery@digitindex by 1\relax
1846 \edef\@texosquery@digit{%
1847   \ifcase\@texosquery@digitindex
1848    0%
1849   \or
1850     \expandafter\@texosquery@firstoften\@texosquery@current
1851   \or
1852     \expandafter\@texosquery@secondoften\@texosquery@current
1853   \or
1854     \expandafter\@texosquery@thirdoften\@texosquery@current
1855   \or
1856     \expandafter\@texosquery@fourthoften\@texosquery@current
1857   \or
1858     \expandafter\@texosquery@fifthoften\@texosquery@current
1859   \or
1860     \expandafter\@texosquery@sixthoften\@texosquery@current
1861   \or
1862     \expandafter\@texosquery@seventhoften\@texosquery@current
1863   \or
1864     \expandafter\@texosquery@eighthoften\@texosquery@current
1865   \or
1866     \expandafter\@texosquery@ninthoften\@texosquery@current
1867   \or
1868     \expandafter\@texosquery@tenthoften\@texosquery@current
1869   \else
1870    0%
1871 \fi
1872 }%
1873 \ifnum\@texosquery@digit=0\relax
1874   \@texosquery@zerodigit
1875 \else
```

```
1876    \if@texosquery@digitfound
1877    \else
1878      \ifx\@texosquery@currentsign\texosquerypatfmtminus
1879       \texosquerypatfmtminus
1880       \let\@texosquery@currentsign\empty
1881      \else
1882        \@texosquery@patfmt@dosep
1883        \let\@texosquery@patfmt@dosep\empty
1884      \fi
1885    \fi
1886    \@texosquery@digitfoundtrue
1887    \@texosquery@digit
1888 \fi
1889 }
```

`\@texosquery@zerodigit@leading`

```
1890 \def\@texosquery@zerodigit@leading{%
1891 \edef\@texosquery@digit{%
1892    \ifcase\@texosquery@digitindex
1893     0%
1894    \or
1895       \expandafter\@texosquery@firstoften\@texosquery@current
1896    \or
1897       \expandafter\@texosquery@firsttwooften\@texosquery@current
1898    \or
1899       \expandafter\@texosquery@firstthreeoften\@texosquery@current
1900    \or
1901       \expandafter\@texosquery@firstfouroften\@texosquery@current
1902    \or
1903       \expandafter\@texosquery@firstfiveoften\@texosquery@current
1904    \or
1905       \expandafter\@texosquery@firstsixoften\@texosquery@current
1906    \or
1907       \expandafter\@texosquery@firstsevenoften\@texosquery@current
1908    \or
1909       \expandafter\@texosquery@firsteightoften\@texosquery@current
1910    \or
1911       \expandafter\@texosquery@firstnineoften\@texosquery@current
1912    \or
1913     \@texosquery@current
1914    \else
1915     0%
1916    \fi
1917    }%
1918 \ifnum\@texosquery@digit>0\relax
1919 \if@texosquery@digitfound
1920 \else
1921     \ifx\texosquerypatminus\texosquerypatfmtminus
1922      \texosquerypatfmtminus
1923     \else
```

107

```
1924        \@texosquery@patfmt@dosep
1925        \let\@texosquery@patfmt@dosep\empty
1926      \fi
1927    \fi
1928    \@texosquery@digitfoundtrue
1929    0%
1930  \fi
1931 }
```

```
1932 \def\@texosquery@zerodigit@trailing{%
1933  \edef\@texosquery@digit{%
1934    \ifcase\@texosquery@digitindex
1935     0%
1936    \or
1937      \@texosquery@current
1938    \or
1939      \expandafter\@texosquery@lastnineoften\@texosquery@current
1940    \or
1941      \expandafter\@texosquery@lasteightoften\@texosquery@current
1942    \or
1943      \expandafter\@texosquery@lastsevenoften\@texosquery@current
1944    \or
1945      \expandafter\@texosquery@lastsixoften\@texosquery@current
1946    \or
1947      \expandafter\@texosquery@lastfiveoften\@texosquery@current
1948    \or
1949      \expandafter\@texosquery@lastfouroften\@texosquery@current
1950    \or
1951      \expandafter\@texosquery@lastthreeoften\@texosquery@current
1952    \or
1953      \expandafter\@texosquery@lasttwooften\@texosquery@current
1954    \or
1955      \expandafter\@texosquery@tenthoften\@texosquery@current
1956    \else
1957     0%
1958    \fi
1959    }%
1960  \ifnum\@texosquery@digit>0\relax
1961    \if@texosquery@digitfound
1962    \else
1963      \ifx\texosquerypatminus\texosquerypatfmtminus
1964       \texosquerypatfmtminus
1965      \else
1966        \@texosquery@patfmt@dosep
1967        \let\@texosquery@patfmt@dosep\empty
1968      \fi
1969    \fi
1970    \@texosquery@digitfoundtrue
1971    0%
```

```
1972   \fi
1973 }
```

\texosquerypatfmtminus   Formatted minus sign. Redefined as appropriate.

```
1974 \def\texosquerypatfmtminus{\ifmmode-\else$-$\fi}
```

\texosquerypatfmtplus   Formatted plus sign. Redefined as appropriate.

```
1975 \def\texosquerypatfmtplus{\ifmmode+\else$+$\fi}
```

\texosquerypatfmtgroupsep

```
1976 \def\texosquerypatfmtgroupsep{,}
```

\texosquerypatfmt@groupsep

```
1977 \def\texosquerypatfmt@groupsep{%
1978   \if@texosquery@digitfound\texosquerypatfmtgroupsep\fi}
```

\texosquerypatfmtpercentsign   Percent sign used in number format.

```
1979 \def\texosquerypatfmtpercentsign{\%}
```

\texosquerypatfmtpermillsign   Per-mill sign used in number format. Redefine as appropriate.

```
1980 \def\texosquerypatfmtpermillsign{‰}
```

\@texosquery@adjust@per

```
1981 \def\@texosquery@adjust@per#1{%
1982 \@texosquery@pos@shift{#1}%
1983 \edef\@texosquery@int{\expandafter\@texosquery@paddigits
1984     \expandafter{\number\@texosquery@int}}%
1985 \edef\@texosquery@frac{\@texosquery@paddigits@trailing{\@texosquery@frac}}%
1986 \edef\@texosquery@current{\expandafter\@texosquery@gobble\@texosquery@int}%
1987 }
```

texosquery@patfmt@prefixpercent

```
1988 \def\texosquery@patfmt@prefixpercent#1#2{%
1989 \@texosquery@adjust@per{2}%
1990   #2\texosquerypatfmtpercentsign#1%
1991 }
```

texosquery@patfmt@suffixpercent

```
1992 \def\texosquery@patfmt@suffixpercent#1#2{%
1993 \@texosquery@adjust@per{2}%
1994 #1\texosquerypatfmtpercentsign#2%
1995 }
```

texosquery@patfmt@prefixpermill

```
1996 \def\texosquery@patfmt@prefixpermill#1#2{%
1997 \@texosquery@adjust@per{3}%
1998 #2\texosquerypatfmtpermillsign#1%
1999 }
```

```
2000 \def\texosquery@patfmt@suffixpermill#1#2{%
2001  \@texosquery@adjust@per{3}%
2002  #1\texosquerypatfmtpermillsign#2%
2003 }
```

All done. Restore the category code of @:

```
2004 \@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:

```
2005 \NeedsTeXFormat{LaTeX2e}
2006 \ProvidesPackage{texosquery}[2020/02/04 v1.7 (NLCT)]
```

Load `texosquery.tex`:

```
2007 \input{texosquery}
```

3.3 Configuration File (`texosquery.cfg`)

The configuration file. This will need to be edited as appropriate to the system. Since Java 7 end of life has long passed, this now defaults to Java 8.

```
2008 %<<COMMENT
2009 % This configuration file must be edited to match your system.
2010 % Copy the file to your TEXMFLOCAL or TEXMFHOME tree to prevent it
2011 % from being overwritten by updates.
2012
2013 % 1. Check your version of Java. To do this run
2014 %      java -version
2015 % from your command line.
2016
2017 %  * If the version number starts with "1.5" or "1.6" then
2018 %    \TeXOSInvokerName must be texosquery-jre5
2019 %    and \TeXOSQueryAllowRestricted must be commented out
2020 %
2021 %  * If the version number starts with "1.7" then
2022 %    \TeXOSInvokerName must be texosquery
2023 %    and \TeXOSQueryAllowRestricted must be commented out
2024 %
2025 %  * If the version number starts with "1.8" or above then
2026 %    \TeXOSInvokerName must be texosquery-jre8
2027 %    and \TeXOSQueryAllowRestricted should be uncommented if
2028 %    texosquery-jre8 is on the restricted list. (This has been the
2029 %    case since TeX Live 2017.)
2030
2031 % (bash users need to check that the .sh extension has been removed
2032 % from the bash scripts.)
```

110

```
2033
2034 \def\TeXOSInvokerName{texosquery-jre8}
2035
2036 % 2. If the invoker name given in the definition above is on the
2037 % restricted list, the line below should be uncommented to allow it to be run
2038 % in restricted mode. Otherwise comment it out.
2039
2040 \TeXOSQueryAllowRestricted
2041
2042 %COMMENT
```

3.4   Bash Scripts

These are the bash scripts for Unix-like systems. The first line

```
#!/bin/sh
```

is added when the files are extracted by `texosquery.ins` (since `\nopreamble` automatically inserts a blank line at the start of the file).

3.4.1   `texosquery.sh`

Adapted from tlcockpit.sh to ensure the script works with cygwin.

```
2043 scriptname=`basename "$0" .sh`
2044 jar="$scriptname.jar"
2045 jarpath=`kpsewhich --progname="$scriptname" --format=texmfscripts "$jar"`
2046
2047 kernel=`uname -s 2>/dev/null`
2048 if echo "$kernel" | grep CYGWIN >/dev/null; then
2049   CYGWIN_ROOT=`cygpath -w /`
2050   export CYGWIN_ROOT
2051   jarpath=`cygpath -w "$jarpath"`
2052 fi
2053
2054 java -jar "$jarpath" "$@"
```

3.4.2   `texosquery-jre8.sh`

```
2055 scriptname=`basename "$0" .sh`
2056 jar="$scriptname.jar"
2057 jarpath=`kpsewhich --progname="$scriptname" --format=texmfscripts "$jar"`
2058
2059 kernel=`uname -s 2>/dev/null`
2060 if echo "$kernel" | grep CYGWIN >/dev/null; then
2061   CYGWIN_ROOT=`cygpath -w /`
2062   export CYGWIN_ROOT
2063   jarpath=`cygpath -w "$jarpath"`
2064 fi
2065
2066 java -Djava.locale.providers=CLDR,JRE -jar "$jarpath" "$@"
```

3.4.3 `texosquery-jre5.sh`

```
2067 scriptname=`basename "$0" .sh`
2068 jar="$scriptname.jar"
2069 jarpath=`kpsewhich --progname="$scriptname" --format=texmfscripts "$jar"`
2070
2071 kernel=`uname -s 2>/dev/null`
2072 if echo "$kernel" | grep CYGWIN >/dev/null; then
2073   CYGWIN_ROOT=`cygpath -w /`
2074   export CYGWIN_ROOT
2075   jarpath=`cygpath -w "$jarpath"`
2076 fi
2077
2078 java -jar "$jarpath" "$@"
```

3.5 Windows Batch Scripts

These are the batch scripts for Windows. TEX on Windows doesn't allow the creation of
`.bat` files, so `.ins` file creates these with the extension `.batch` which will need to be
changed to `.bat` after extraction.

3.5.1 `texosquery.bat`

```
2079 @ECHO OFF
2080 FOR /F "tokens=*" %%I IN ('kpsewhich --progname=texosquery --format=texmfscripts texosquery.jar
2081 java -jar "%JARPATH%" %*
```

3.5.2 `texosquery-jre8.bat`

```
2082 @ECHO OFF
2083 FOR /F "tokens=*" %%I IN ('kpsewhich --progname=texosquery --format=texmfscripts texosquery-jre8
2084 java -Djava.locale.providers=CLDR,JRE -jar "%JARPATH%" %*
```

3.5.3 `texosquery-jre5.bat`

```
2085 @ECHO OFF
2086 FOR /F "tokens=*" %%I IN ('kpsewhich --progname=texosquery --format=texmfscripts texosquery-jre5
2087 java -jar "%JARPATH%" %*
```

Abbreviations

ASCII American Standard Code for Information Interchange

BCP Best Common Practice

CLDR Unicode Consortium's Common Locale Data Repository

IETF Internet Engineering Task Force

ISO International Organization for Standardization

JRE Java Runtime Environment

OS operating system

POSIX Portable Operating System Interface

UTF Unicode Transformation Format

Change History

117

119

Index