

latexfileinfo-pkgs — Displaying [Metadata](#) (“Info”) of *LaTeX* Source Files

— a comparison of packages available from CTAN

[[intro](#) | [table](#) | [\GetFileInfo etc.](#) | [\listfiles etc.](#) | [rcs/svn](#) | [thanks](#)] [[reload](#)]

[[⇒](#) | [↑](#)]

0 Introduction (with links to sections)

For [LaTeX](#) source files, it is recommended to start them with LaTeX command

- `\ProvidesClass` (for `.cls` files)
- `\ProvidesPackage` (for `.sty` files)
- `\ProvidesFile` (for all other files)

(after `\NeedsTeXFormat`). Their trailing optional argument can provide the most recent

- [revision](#) **date**
(expected as “*first word*” in trailing optional argument)
- the corresponding **version** string
(expected as “*second word*” in trailing optional argument) — and
- and a brief **description** of the file (“caption”)
(expected as anything *else* in trailing optional argument).

Here we are listing and comparing TeX packages available from [CTAN](#) that make use of this information.

The document especially describes five packages of my own, including interrelations and dependencies, and thus extends their documentations; see [table below](#) for them. An *asterisk** near a package name indicates that the package is *not mine*; so what I tell about it may be wrong or bad ... (please let me know via [form](#), or should I add something?) ...

We have the *cases* of

1. displaying [single](#) file info (`\GetFileInfo ...`)
2. listing infos of [several](#) files (`\listfiles` etc.)

and for each

- a. access by *LaTeX* ([1](#) · [2](#))
- b. access by *external* program ([1](#) · [2](#)).

Related subjects are using

- i. [revision control systems](#) and
- ii. modification time according to [pdfTeX](#).

A *tabular overview* of the first two case distinctions:

	just using <i>LaTeX</i>	using <i>external</i> program
single file	<ul style="list-style-type: none"> • doc* (Frank Mittelbach) • zwgetfdte* (Zdeněk Wagner) • readprov 	<ul style="list-style-type: none"> • latexfileversion* (Harald Harders) • ltxfileinfo* (Wybo Dekker) • typeoutfileinfo
file list	<ul style="list-style-type: none"> • classlist* (Heiko Oberdiek) • dateiliste* (Paul Ebermann) • longnamefilelist • nicefilelist • myfilist 	<ul style="list-style-type: none"> • [ltxinput* (Jim Green)] • [texlog-extract* (Wybo Dekker)] <p>[don’t really belong here ...]</p>

[\[⇒|↑\]](#)

1 Access Single File Info

[[a: doc](#) | [readprov](#) | [zwgetfdte](#)] [[b: latexfileversion](#) | [ltxfileinfo](#) | [typeoutfileinfo](#)] [[c: filemod](#)]

a. Access in *LaTeX* Run

—originally for *automatic* inclusion of *current* revision date and version of a package in *typesetting* the latter’s documentation (“*This document describes version ... as of ...*”), but then ...

[doc](#)*

- Belongs to LaTeX [base](#) distribution.
- Provides `\GetFileInfo{<file>}`; after using this, you have `\filedate`, `\fileversion`, `\fileinfo` of `<file>`.
- `<file>` must have been *loaded* in the LaTeX run.
- `\GetFileInfo` is *fragile*.

[zwgetfdte](#)*

- Provides `\DateOfFile{<file>}` and `\DateOfPackage{<file>}`.
- `<file>` must have been *loaded*.

[readprov](#)

- Provides `\UseDateOf{<file>}` and `\UseVersionOf{<file>}` – *robust* (expandable).
- Provides [doc](#)’s `\GetFileInfo` (with modified definition; used internally)
- Provides
 - `\ReadFileInfos{<files>}`
 - `\ReadPackageInfos{<files>}`
 - `\ReadClassInfo{<file>}`
 as *alternatives* to `\GetFileInfo`.
 - They do *not* really *load* `<file>/<files>` ...
 - ... so can be used with incompatible packages and classes (multiple latter by `\ReadFileInfos{<base>.cls}`).
 - This allows **adding arbitrary** LaTeX source files to the list of files that LaTeX manages for [\listfiles](#) — this is used for [myfilist](#).

b. Screen Display Using *External* Program

—when you wonder whether the most recent version of a package or a chapter file is present ...

[latexfileversion](#)*

- [Bash \(Unix shell\)](#) script.
- *Nice* screen display.
- Invokes latex run using a copy of [doc](#)'s `\GetFileInfo`.

[ltxfileinfo](#)*

- [Ruby](#) script.
- Displays nice [table](#) of infos (including *location* in file system) on screen.

[typeoutfileinfo](#)

- 5-lines [Bash \(Unix shell\)](#) script.
- Expects latex won't append too many screen lines after 1-line `\typeout` (*tolerable* screen display as compared with [latexfileversion](#)).
- Invokes latex run using [readprov](#).

c. Related, while different ...

[filemod](#)*

- Uses [pdfTeX](#)'s `\pdffilemoddate{<file>}` primitive.
- Compares modification dates of files.
- Files include images.
- Also works with pdf`latex` in DVI mode.

[\[⇒|↑\]](#)

2 File Info List

This is about listing *all* files input in a LaTeX run, *or* ...

a. Create in LaTeX Run

[[LaTeX](#) | [classlist](#) | [dateiliste](#) | [longnamefilelist](#) | [myfilist](#) | [nicefilelist](#)]

[latex](#)*

- LaTeX provides `\listfiles` for the document preamble.
- `\listfiles` issues a list of **all** files input in the latex run near end of .log file, together with their infos according to `\Provide...` commands.
- The list is a **two-column** “table” (plain text), *base* filenames flush right, *info* (maybe date, maybe version, maybe anything) flush left.
- Any file whose base [filename](#) has **more than 8** characters or whose filename extension does *not* have 3 characters **corrupts** alignment.

[classlist](#)*

- Remembers (separately) files input by (a) `\documentclass` and what was input by (b) `\LoadClass`.
- On `\PrintClassList`, the list of loaded class files appears on screen.

- Configurable by `\PrintClassListEntry` and `\PrintClassListTitle`.

[dateiliste](#)*

- Lists anything that the [original](#) `\listfiles` would list.
- **Typesets** the list as a LaTeX table, using package [longtable](#).
- Separate **columns** for
 - filename
 - *page* where input (optionally)
 - date
 - version
 - description (“caption”)
- Supports **RCS**, [CVS](#), and **SVN** (cf. [section below](#)).
- Highly configurable.
- E.g., you can replace info for a file by what you want to see in the list (cf. [myfilist](#)).
- I haven’t seen what happens with *plain text* (.log) output.

[longnamefilelist](#)

- Overcomes [LaTeX](#)’s “8-character limit” by a new **optional argument** for `\listfiles`, indicating number of characters to be reserved for base filenames, e.g. accounting for “longnamefilelist.sty”:

```
\listfiles[16]
```

— view [example outcome](#)

- —showing combination with [myfilist](#)
- —while also working as single addition to LaTeX [base](#), *no* need of [myfilist](#) or [monofill](#).
- Still, filename **extensions** with number of characters differing from 3 break alignment. Martin Münch mentions `t1cmtt.fd` (standard LaTeX font definition) and `supp-pdf.mkii` ([ConTeXt](#)).

[nicefilelist](#)

- Uses separate **columns** for **date** and **time** (like [dateiliste](#)).
- (Picky) recognition of “date” and “version” — if not present, left **empty** or gets “missing” display:

```
nicefilelist.sty  2012/03/29  v0.2  more file list alignment (UL)
nicefilelist.tex  2012/03/23  --    documenting nicefilelist.sty
```

(Martin Münch’s idea).

- Problem of varying filename **extension** lengths overcome by actually keeping a separate flush-left column for them.
- Column widths configurable by **templates** — replace pre-configuration ([monofill](#) commands):

```
\MFfieldtemplate{f-base}    {nicefilelist}  %% base name
\MFfieldtemplate{f-ext}     {tex}            %% name extension
\MFfieldtemplate{f-version}{v0.11a}         %% version
```

- First code line above shows **difference** to [longnamefilelist](#), equivalent there would be `\listfiles[12]`. `nicefilelist` does *not* provide an optional argument for `\listfiles`.
- Column **distances** and “**missing**” display configurable too.
- Can be combined with [myfilist](#) — view [output sample](#).
- Needs just [monofill](#), additionally to LaTeX [base](#).
- `nicefilelist` v0.4 provides an option `[r]` to allow “release numbers” in the column reserved for versions:

```
nicefilelist.sty  2012/05/20  v0.4  more file list alignment (UL)
nicefilelist.tex  2012/05/20  --    documenting nicefilelist.sty
nicefilelist.RLS  2012/05/20  r0.4  v0.4 Kabelschacht + [r]
```

.RLS files are an idea to provide/access a release summary.

[myfilist](#)

- Allows **removing** all entries that LaTeX has collected for [\listfiles](#) at a certain point (`\EmptyFileList`).
- Then, by commands from [readprov](#), you can **add arbitrary** files to the list (as they are *not* really *loaded*), in the order you want to have them in the list. (However, for many combinations of package files, you could actually *load* them, without `readprov`.)
- Thus actually, you can set up a list of files you want to have *independently* of a typesetting run ...
- ... and actually, the **intended** application is generating the list **without** any **typesetting**, by running `latex` on a “**script**” file just loading `readprov` and `myfilist` and using only commands from *them*. (No `\documentclass`, no `{document}` environment.)
- View [input example](#) for [longnamefilelist](#).
- Besides .log output, the list can be written into a **separate** plain text **file** — view [output example](#) for [longnamefilelist](#).
- **Main applications** in my mind and actual work:
 1. List of .sty and .tex files for *my* **CTAN packages** and **bundles** (instead of the .dtx/[docstrip](#) system, I use [nicetext](#)).
 2. List of **package** files specific to some **project** that often change — manual replacement for a “concurrent versions” system with single author. E.g.:
 - [nicetext](#) packages underlying **documentation** of a certain different package.
 - **style** files underlying a **book** project at which authors and programmers work on changing computers.
 3. List of **chapter** `\include` files for a **book** project, edited at changing computers (by a number of authors).
 4. Often, one **forgets** to **update** version information in the `\Provides...` command — the `myfilist` helps you to check this (e.g., right before a release; [filemod](#) might automate this check, perhaps together with [readprov](#), *without* `myfilist`).
- My actual **workflow** with [Bash \(Unix shell\)](#):
 - **upsf1** with


```
alias upsf1='latex srcfiles'
```

updates the list of source files, and actually displays the updated list on screen.
 - **shsf1** with


```
alias shsf1='more SrcFILES.txt'
```

displays the source file list without updating, and waits when the list is too long for your (netbook) screen.
- Combinable with [dateiliste](#)?

b. Create by External Program

Hm, not so related, *no* infos (it seems), lists only; so, just “*most related I could find*” for this section ...

[Itxinput](#)*

- [MS-DOS](#) utility (compiled from [C](#)).
- Lists files that *would* be input (recursively).

[texlog-extract](#)*

- [Ruby](#) script.
- Lists files that issued errors and warnings (colored).

3 RCS/SVN

For [revision control systems RCS](#) and [SVN](#), there are

- [rcs*](#) · [rcsinfo*](#) · [rcs-multi*](#)
- [svn*](#) · [svninfo*](#) · [svn-multi*](#)
- [svn-prov*](#) provides SVN variants of `\ProvidesClass`, `\ProvidesFile`, and `\ProvidesPackage`.

—sorry, I can't tell more about them right now ...

Acknowledgements

Thanks to Martin Münch, Moss (I wrote this for his question), and Rainer Schöpf! [RCS stuff](#) mainly has been stolen from [Jürgen Fenn's Topic Index](#) of the [TeX Catalogue](#).

* not mine

Last revised 2012-05-29 © [Uwe Lück](#)

(using [blog.sty](#) and [monofill.sty](#))

License: [LPPL 1.3c](#) or later, author-maintained.

[\[⇒ top of page\]](#)

