

myfilist

List Infos on just the Files You Want to Know About*

Uwe Lück[†]

October 25, 2012

Abstract

`myfilist` addresses lazy file versions management, when you move your package or chapter files through various computers and various directories and after a while wonder where the most recent versions are.

Like Paul Ebermann's `dateiliste`,¹ `myfilist` varies L^AT_EX's `\listfiles` for listing file (especially version) informations. Differences to `dateiliste` and L^AT_EX are:

1. You choose the files (and their order) to be listed;
2. indeed: this has very little to do with files used in typesetting some document;
3. output is just screen, `.log`, or a `.txt`-type file that you choose.

The code of this package was ready in spring 2008; in spring 2010 I presented it as a kind of study on improving `nicetext`'s² `txt-to-LATEX` processing after `nicetext v0.4` (aim was to do this without modifying the documented file, yet I did not obey this ruly strictly here). The new idea is adding `wiki.sty`'s “environments” feature to `makedoc` and `niceverb`'s “auto mode” in order to interpret `txt` comment indents in the package file, while `wiki.sty`'s `font` switching still is not compatible with `niceverb`'s. Easy script commands for achieving this are still missing (sorry; see the code in `myfilist.tex` that achieved the present formatting.)

*This file describes version `v0.7` of `myfilist.sty` as of 2012/10/25.

[†]<http://contact-ednotes.sty.de.vu>

¹<http://ctan.org/pkg/dateiliste>

²<http://ctan.org/pkg/nicetext>

Contents

1	Installing	2
2	File Info Header	2
3	Usage	3
4	Variants	4
5	Tricks, Package Option	4
6	Implementation	5
7	Example	8

1 Installing

The file `myfilist.sty` is provided ready, installation only requires putting it somewhere where \TeX finds it (which may need updating the filename data base).³

2 File Info Header

```

1  \def \filename {myfilist.sty} %% macro package for LaTeX,
2
3  %% modifies \listfiles to choose files to be listed and
4  %% list them in a chosen file.
5
6  \def \fileversion {0.7}      \def \filedate {2012/10/25}
7
8  %% copyright (C) 2008, 2010, 2011, 2012 Uwe Lueck,
9  %% http://www.contact-ednotes.sty.de.vu
10 %% -- author-maintained in the sense of LPPL below.
11 %%
12 %% This file can be redistributed and/or modified under
13 %% the terms of the LaTeX Project Public License; either
14 %% version 1.3a of the License, or any later version.
15 %% The latest version of this license is in
16 %%      http://www.latex-project.org/lppl.txt
17 %% We did our best to help you, but there is NO WARRANTY.
18 %%
19 %% Please report bugs, problems, and suggestions via
20 %%
21 %%      http://www.contact-ednotes.sty.de.vu
22 %%
```

³<http://www.tex.ac.uk/cgi-bin/texfaq2html?label=inst-wlcf>

3 Usage

Write a script file like `gather.tex` which you should have received together with `myfilist.sty`. We use L^AT_EX 2_ε (required!), yet (in general) not for typesetting a document, therefore the script file needn't contain `\documentclass`. `myfilist.sty` is loaded by

```
\RequirePackage{myfilist}
```

instead of `\usepackage{myfilist}`. Usually you don't want to have `myfilist.sty` in the list, so type

```
\EmptyFileList
```

next; or type it after some additional `\RequirePackage...` for packages you want to use but not list.

Next list the names of the files whose informations you want to get as arguments of

```
\ReadFileInfos or \ReadPackageInfos or \ReadClassInfo
```

With the latter two, you can omit “.sty” or “.cls” as with `\usepackage` and `\documentclass`. With the first one, the file name extension “.tex” may be omitted. The two first commands accept lists with commas as separators almost like with `\usepackage` (currently we must use “%” to hide a line break in the script, and there must be no spaces in the list).—For more details on these commands, please see `readprov.pdf`.

The list of files is actually printed in the transcript `.log` and on screen on

```
\ListInfos
```

The list can additionally be written into a plain text file,

```
\ListInfos[filelist.txt]
```

as an example; i.e., you can use the brackets to tell which file is to contain your list of files. The file informations appear in the order of their names in your script file.

`\ListInfos` concludes, ends the job, anything following in the script file will be ignored.—All of this requires that you *run* `gather.tex` or your other script file, about as

```
latex gather
```

To list files that were loaded earlier in a different order, list them in an *optional argument* of `\EmptyFileList`, e.g.,

```
\EmptyFileList[myfilist.sty,readprov.sty]
```

Note: `myfilist.sty` (in its present state) is unable to list files that don't contain `\ProvidesPackage`, `\ProvidesClass`, or `\ProvidesFile`. `myfilist.sty` “loads”

files to extract version informations, but quits them as soon as it has found one of these $\text{\LaTeX 2}_{\epsilon}$ commands.

4 Variants

v0.5 provides different interfaces to reduce (shared) \TeX code when (similar) shell or batch scripts generate that `gather.tex`.

```
\ReadListFileInfos[<txt-file>]{<tex-files>}
```

is a shorthand for

```
\ReadFileInfos{<tex-files>}
\ListInfos[<txt-file>]
```

Without the optional argument, the screen output is *not* written to disk—by default. However, preceding `\ReadListFileInfos` by

```
\WriteFileInfosTo{<txt-file>}
```

is an alternative way to get the list in the plain text file.

```
\FindReadListFileInfos[<txt-file>]{<tex-files>}
```

must be used rather than `\ReadListFileInfos` when the comma-separated list (generated by the “`find`” utility) *starts* with a comma. Alternatively,

```
\UseFindUtility
```

preceding `\ReadListFileInfos` deals with the leading comma.

v0.7 adds

```
\NoStopListInfos[filelist.txt]
```

after which the script can continue until `\stop`, e.g., for checks with `filedate.sty` so that the latter’s messages are the final lines you see on screen.

5 Tricks, Package Option

v0.6 first provides (e.g.)

```
\FileListRemark[----]{---DOC.---}
```

in order to insert `---DOC.---□□---` in the list. This writes funny files that may be somewhat dangerous ... I have used a bash script version of this for structuring large file lists.

v0.6 moreover provides two hooks and a package option useful for the `adhocfilelist` package, or perhaps even otherwise.

`\ListGenerator`

is a hook for the final lines in the plain text output file (see implementation).

`\NoBottomLines`

suppresses final messages of the L^AT_EX run, which may be nicer and more informative on the screen. Package Option

`[no-bot]`

is a kind of shorthand to issue `\NoBottomLines`.

6 Implementation

```

23 \NeedsTeXFormat{LaTeX2e}[1994/12/01] %% \newcommand* etc.
24 \ProvidesPackage{myfilist}
25     [\filedate\space v\fileversion \space
26     \string\listfiles\space-- mine only (UL)]

\ReadFileInfos, \ReadPackageInfos, and \ReadClassInfo are imple-
mented in readprov.sty:

27 \RequirePackage{readprov}[2010/11/26]

\EmptyFileList[\read-again-files]

28 \newcommand{\EmptyFileList}[1][]{%
29     \let\@filelist\@gobble
30     \@for\@tempa:=#1\do{%
31         \global \expandafter \let \csname ver@\@tempa\endcsname \relax
32         %% v0.4:
33         \global \expandafter \let \csname opt@\@tempa\endcsname \relax}}

\ListInfos[\output-file]

34 \newcommand*\ListInfos[1][]{%
35     \ifx$#1$\@dofilelist\else
36         % \newwrite\file@of@filelist           %% mv. v0.6
37         \immediate\openout\file@of@filelist=#1\relax
38         \def\typeout##1{%
39             \read@\typeout{##1}%
40             \immediate\write\file@of@filelist{##1}}%
41         \@dofilelist
42         %% made at:
43         \@tempcnta\time      \@tempcntb\@tempcnta
44         \divide\@tempcnta 60 \count@\@tempcnta
45         \multiply\count@ 60 \advance\@tempcntb -\count@
46         \typeout{ List made at
47             \the\year/\two@digits{\the\month}/\two@digits{\the\day},
48             \two@digits{\the\@tempcnta}:\two@digits{\the\@tempcntb}%

```

v0.6 becomes more flexible here:

```

49      \ListGenerator}
50      \immediate\closeout\file@of@filelist
51      \fi
52      \stop}
53      %% <- TODO how to suppress first empty line? 2008/03/16
54      \newwrite\file@of@filelist          %% mv. v0.6

```

emulating versions before v0.6:

```

55      \providecommand*\ListGenerator{%
56          ^^J from script file \jobname.tex^^J}
57      \let\read@typeout\typeout

```

`\NoStopListInfos[⟨output-file⟩]` (v0.7)

```

58      \newcommand*\NoStopListInfos[1][]{\let\stop\relax \ListInfos[#1]}

```

v0.5:

There is a little (design) bug in `\ListInfos`: When `gather.tex`'s last line is `\ListInfos` without optional argument or anything, \LaTeX prompts for further input. This might be fixed using `ifnextok.sty`, but I am not sure right now. In `\VarListInfos`, `\relax` stops waiting for input. `\providecommand` means that the default can be overridden before loading `myfilist.sty`:

```

59      \providecommand*\VarListInfos{\ListInfos\relax}

\WriteFileInfosTo{⟨txt-file⟩} overrides the previous default:

60      \newcommand*\WriteFileInfosTo[1]{%
61          \renewcommand\VarListInfos{\ListInfos[#1]}

\ReadListFileInfos[⟨txt-file⟩]{⟨tex-files⟩}

62      \newcommand*\ReadListFileInfos[2][]{%
63          \ifx$#1$\else\WriteFileInfosTo{#1}\fi
64          \read@comma@guard\read@list@file@infos#2\stop}
65      \let\read@comma@guard\empty
66      \def\read@list@file@infos#1\stop{%

```

Here `\VarListInfos` is used finally:

```

67      \ReadFileInfos{#1}\VarListInfos}

\UseFindUtility activates \read@comma@guard in order to kill a leading
comma:

```

```

68      \newcommand*\UseFindUtility{%
69          \let\read@comma@guard\@firstoftwo}

\FindReadListFileInfos[⟨txt-file⟩]{⟨tex-files⟩}

70      \newcommand*\FindReadListFileInfos{%
71          \UseFindUtility\ReadListFileInfos}

```

v0.6:

`\FileListRemark[⟨info⟩]{⟨filename⟩}` \writes and reads a dummy file
 ⟨filename⟩ with \Provides... content ⟨info⟩ which is empty by default:

```

72 \newcommand*\FileListRemark[2][\{%
73   \immediate\openout\file@of@filelist#2
74   \immediate\write\file@of@filelist{%
75     \string\ProvidesFile{#2}[#1]}%
76   \immediate\closeout\file@of@filelist
77   \ReadFileInfos{#2}}

```

`\NoBottomLines`

```

78 \newcommand*{\NoBottomLines}{%
79   \let\STOP\stop \def\stop{\batchmode\STOP}}

```

Package option `[no-bot]`:

```

80 \DeclareOption{no-bot}{\NoBottomLines}
81 \ProcessOptions

```

Finally issuing `\listfiles` (already before v0.5):

```

82 \listfiles
83
84 \endinput
85
86 %% VERSION HISTORY
87 v0.1    2008/03/19  created file ‘readprov.sty’
88          2008/03/23  smart file name separation, \ifx$ for \ifcat$
89          2008/05/22  typo ist -> it
90 v0.2    2010/04/03  renamed ‘myfiles.sty’;
91          broke long lines etc. for doc
92 SENT TO CTAN -- declined
93 v0.2a   2010/04/08  undoubled \NeedsTeXFormat, copyright updated
94 v0.3    2010/11/26  split off from ‘myfiles.sty’, opt. arg. for
95                  \EmptyFileList, doc. reworked
96          2010/11/27  some USAGE clarifications
97 v0.3a   2011/01/29  two empty lines commented out
98          2011/01/30  copyright updated
99 v0.4    2012/05/26  \EmptyFileList clears options
100 v0.5    2012/09/27  \ReadListFileInfos, \WriteFileList,
101                  \FindReadListInfos, \UseFindUtility
102 v0.6    2012/10/11  \FileListRemark, \ListGenerator,
103                  \NoBottomLines, [no-bot]
104 v0.6a   2012/10/12  adding \UseFindUtility and \FindReadListInfos
105                  in the history for v0.5
106 v0.7    2012/10/25  \NoStopListInfos
107

```

7 Example

`gather.tex` for the present bundle has been as follows (added development versions of `fifinddo.sty` etc.):

```
\ProvidesFile{gather.tex}[2012/10/25 collect file infos]
\RequirePackage[no-bot]{myfilist}
\EmptyFileList[readprov.sty,myfilist.sty]
\FileListRemark[-----]{-RELEASE.---}
\ReadFileInfos{fileinfo.RLS}
\FileListRemark[-----]{-----USE.---}
\ReadPackageInfos{readprov,myfilist}
\FileListRemark[-----]{-----DOC.---}
\ReadFileInfos{readprov,myfilist}
\FileListRemark[-----]{-----USED.---}
\ReadFileInfos{README.tex,makedoc.cfg}
\ReadPackageInfos{fifinddo,makedoc,niceverb}
\ReadFileInfos{gather}
\NoStopListInfos[CONTENTS.txt]
\RequirePackage{dowith,filedate}
\ModDates
\UseReferenceDate{\thepdfmoddate}
\DoWithAllOf\CheckDateOfGiven{{fileinfo.RLS}
    {readprov.sty}{myfilist.sty}
    {readprov.tex}{myfilist.tex}
    {gather.tex}}
\stop
```