

commado.sty and filesdo.sty

Immediately Extend a One-Argument Macro to Comma-Separated Lists and Combinations of Filename Bases and Extensions*

Uwe Lück[†]

November 30, 2012

Abstract

`commado.sty` provides

```
\DoWithCSL{<cmd>}{<list>}
```

in order to apply an existing one-parameter macro $\langle cmd \rangle$ to each item in a list $\langle list \rangle$ in which items are separated by commas. `filesdo.sty` provides

```
\DoWithBasesExts{<cmd>}{<bases>}{<exts>}
```

in order to run $\langle cmd \rangle\{\langle base \rangle.\langle ext \rangle\}$ for some (at most) one-parameter macro $\langle cmd \rangle$, each base filename $\langle base \rangle$ in the comma-separated list $\langle bases \rangle$ and each filename extension $\langle ext \rangle$ in the comma-separated list $\langle exts \rangle$. As opposed to L^AT_EX's internal `\@for`, no assignments are involved (unless $\langle cmd \rangle$ uses assignments—"expandability" in "T_EX's gullet").

Both packages are "generic," i.e., should work with Plain T_EX, L^AT_EX or even other formats, relying on the `plainpkg` package for some minimal L^AT_EX-like behaviour.

Related packages: `loops` and others mentioned in the `dowith` package documentation.

Keywords: macro programming, programming structures, loops, lists

*This document describes v0.11 of `commado.sty` as of 2012/11/30 and version v0.1 as of 2012/11/27 of `filesdo.sty`.

[†]<http://contact-ednotes.sty.de.vu>

Contents

1	Installing and Calling	2
2	Syntax and Relation to the <code>dowith</code> Package	3
3	Example for <code>filesdo.sty</code>	3
4	The File <code>commado.sty</code>	4
4.1	Package File Header (Legalese and <code>plainpkg</code>)	4
4.2	Auxiliaries	4
4.3	Processing a Comma-Separated List	4
4.4	Leaving the Package File	5
4.5	VERSION HISTORY	5
5	The File <code>filesdo.sty</code>	6
5.1	Package File Header (Legalese and <code>plainpkg</code>)	6
5.2	Documentation	6
5.3	<code>commado</code> Required	6
5.4	Category Code	6
5.5	User Commands	6
5.6	Leaving the Package File	7
5.7	VERSION HISTORY	7

1 Installing and Calling

The files `commado.sty` and `filesdo.sty` are provided ready, installation just requires putting them somewhere where $\text{T}_{\text{E}}\text{X}$ finds them (which may need updating the filename data base).¹ However, installation of the package `plainpkg`² and the package `stacklet.sty` (`catcodes`³ bundle) is required additionally.

As to calling (loading): `commado.sty` and `filesdo.sty` are “`plainpkg` packages” in the sense of the `plainpkg` documentation that you may consult for details. So roughly,

- load it by `\usepackage{<pkg>}` if you can,
- otherwise by `\RequirePackage{<pkg>}`
(perhaps from within another “`plainpkg` package”),
- or by `\input_<pkg>.sty`
- or even by `\input{<pkg>.sty}` . . .

—where `<pkg>` is ‘`commado`’ or ‘`filesdo`’.

¹<http://www.tex.ac.uk/cgi-bin/texfaq2html?label=inst-wlcf>

²<http://ctan.org/pkg/plainpkg>

³<http://ctan.org/pkg/catcodes>

2 Syntax and Relation to the `dowith` Package

In $\langle list \rangle$ with `\DoWithCSL{ $\langle cmd \rangle$ }{ $\langle list \rangle$ }`, blank spaces before entries or after commas as well as preceding the closing brace are ignored. So

```
\DoWithCSL{ $\langle cmd \rangle$ }{ $\langle cfg \rangle$ , $\langle sty \rangle$ , $\langle tex \rangle$ }
```

works like

```
\DoWith{ $\langle cmd \rangle$ }{ $\langle cfg \rangle$ }{ $\langle sty \rangle$ }{ $\langle tex \rangle$ }
```

from `dowith.sty`.⁴ With `\DoWithCSL` (at present), an item cannot be empty or consist of blank spaces only. Empty or blank items can be handled by `\DoWith`.

3 Example for `filesdo.sty`

In the file `srcfiles.tex` for the `nicetext`⁵ bundle, there is a line

```
\DoWithBasesExts\ReadFileInfos{fifinddo,niceverb}{sty,tex}
```

This works like

```
\ReadFileInfos{fifinddo.sty}
\ReadFileInfos{niceverb.sty}
\ReadFileInfos{fifinddo.tex}
\ReadFileInfos{niceverb.tex}
```

or actually (a special feature of `readprov`'s⁶ `\ReadFileInfos` is that its argument may be a comma-separated list already)

```
\ReadFileInfos{fifinddo.sty,niceverb.sty,
               fifinddo.tex,niceverb.tex}
```

I ponder providing a shorthand `\ReadBaseExtInfos` for

```
\DoWithBasesExts\ReadFileInfos
```

and reimplementing `\ReadFileInfos` using `\DoWithCLS` in `myfilist.sty` (2012-11-27).

⁴<http://ctan.org/pkg/dowith>

⁵<http://ctan.org/pkg/nicotext>

⁶<http://ctan.org/pkg/readprov>

4 The File `commado.sty`

4.1 Package File Header (Legalese and `plainpkg`)

```

1                                     \input plainpkg
2 % \NeedsTeXFormat{LaTeX2e}[1994/12/01]
3 \ProvidesPackage{commado}[2012/11/30 v0.11 iterate on CSL (UL)]
4 %%
5 %% Copyright (C) 2012 Uwe Lueck,
6 %% http://www.contact-ednotes.sty.de.vu
7 %% -- author-maintained in the sense of LPPL below --
8 %%
9 %% This file can be redistributed and/or modified under
10 %% the terms of the LaTeX Project Public License; either
11 %% version 1.3c of the License, or any later version.
12 %% The latest version of this license is in
13 %% http://www.latex-project.org/lppl.txt
14 %% We did our best to help you, but there is NO WARRANTY.
15 %%
16 %% Please report bugs, problems, and suggestions via
17 %%
18 %% http://www.contact-ednotes.sty.de.vu
19 %%
20 \PushCatMakeLetterAt
```

4.2 Auxiliaries

```

21 \ifltx \else %% unless provided by LaTeX already
22     \long\def\@firstoftwo#1#2{#1}
23     \long\def\@secondoftwo#1#2{#1}
24 \fi

... 4 less than

25 % \long\def\@firstsecondofthree#1#2#3{#1#2}
26 % \long\def\@firstthirdofthree#1#2#3{#1#3}
```

4.3 Processing a Comma-Separated List

Most of the following code aims at removing the final space in the comma-separated list. A variant of parsing as in `fifinddo.sty` (`nicetext`⁷ bundle) and `bitelist.sty`⁸ package is employed (while I am about to use different approaches there, one may see here how, inspired by `\@ifblank` in `url.sty`).⁹ The purpose of the following `\edef` of `\DoWithCSL{<cmd>}{<list>}` is to get a space token after `\@firstoftwo` in the parameter text.

⁷<http://ctan.org/pkg/nicetext>

⁸<http://ctan.org/pkg/bitelist>

⁹<http://ctan.org/pkg/url>

```

27 \let\CD@final@comma\relax
28 \edef\DoWithCSL#1#2{%
29   \CD@final@comma#2\CD@final@comma   %% 2nd \ 2012/11/30
30   \noexpand\@firstoftwo
31   %   \noexpand\@firstsecondofthree
32   \space\CD@final@comma
33   \noexpand\@secondoftwo
34   %   \noexpand\@firstthirdofthree
35   \noexpand\end{#1}{#2}}
36 \def\CD@final@comma#1 \CD@final@comma#2#3\end#4#5{%
37   %   \expandafter\@secondfirstoftwo\expandafter{%
38   %     #2{#1}#5}\do@with@csl#4}%
39   #2{\do@with@csl{#4}#1}\do@with@csl{#4}#5}%

```

... 15 vs. 13:

```

40 %   #2{\do@with@csl{#4}}{#1}{#5}%
41   ,\StopDoing,}
42 \catcode'\Q=3   %% not in #1
43 \def\do@with@csl#1#2#3,{%

```

#1 is *<cmd>*. #2 takes the first token from (remaining) *<list>* that is not a space token. Trying to enter a blank list item would result in using the *next comma as a list item!*—The following is an alternative to the analogue in *domore.sty*:

```

44   \unless@stop@doing#2#3\StopDoing
45   #1{#2#3}\do@with@csl{#1}\StopDoing Q}
46 \def\unless@stop@doing#1\StopDoing#2\StopDoing#3Q{#2}

```

... somewhat replaces `\@secondfirstoftwo`—but will the latter be dropped? `\unless@stop@doing` is specific for `\StopDoing`—but can be used with `\DoWith` too. **TODO**

```

47 \catcode'\Q=11

```

4.4 Leaving the Package File

```

48 \PopLetterCatAt
49 \endinput

```

4.5 VERSION HISTORY

```

50 v0.1   2012/11/24f.  started
51       2012/11/26   code ready
52       2012/11/27   documented
53 v0.11 2012/11/30   code typo corrected, removing final space,
54                          doc. \urlfoot's
55

```

5 The File filesdo.sty

5.1 Package File Header (Legalese and plainpkg)

```

1                                     \input plainpkg
2 % \NeedsTeXFormat{LaTeX2e}[1994/12/01]
3 \ProvidesPackage{filesdo}[2012/11/27 v0.1 iterate on files (UL)]
4 %%
5 %% Copyright (C) 2012 Uwe Lueck,
6 %% http://www.contact-ednotes.sty.de.vu
7 %% -- author-maintained in the sense of LPPL below --
8 %%
9 %% This file can be redistributed and/or modified under
10 %% the terms of the LaTeX Project Public License; either
11 %% version 1.3c of the License, or any later version.
12 %% The latest version of this license is in
13 %%   http://www.latex-project.org/lppl.txt
14 %% We did our best to help you, but there is NO WARRANTY.
15 %%
16 %% Please report bugs, problems, and suggestions via
17 %%
18 %%   http://www.contact-ednotes.sty.de.vu
19 %%

```

5.2 Documentation

For documentation in PDF format, see `commado.pdf`.

5.3 commado Required

filesdo is based on commado:

```
20 \RequirePackage{commado}
```

5.4 Category Code

Use @ as part of “command names” (plainpkg, stacklet):

```
21 \PushCatMakeLetterAt
```

5.5 User Commands

`\DoWithExtBases{<cmd>}{<ext>}{<basenames>}` runs `<cmd>{<base>.<ext>}` for all items `<base>` in `<basenames>` and a single filename extension `<ext>`:

```
22 \def\DoWithExtBases#1#2{\DoWithCSL{\do@with@ext@base{#1}{#2}}}
23 \def\do@with@ext@base#1#2#3{#1{#3.#2}}
```

`\DoWithBasesExts{<cmd>}{<basenames>}{<exts>}` runs `<cmd>{<base>.<ext>}` for all items `<base>` in `<basenames>` and all items `<ext>` in `<exts>`:

```
24 \def\DoWithBasesExts#1#2{\DoWithCSL{\distrib@basenames@do{#1}{#2}}}
    \distrib@basenames@do exchanges arguments in order to reduce the task to
    \DoWithCSL and \DoWithCSL:
25 \def\distrib@basenames@do#1#2#3{%
26     \DoWithCSL{\DoWithExtBases{#1}{#3}}{#2}}
```

5.6 Leaving the Package File

```
27 \PopLetterCatAt
28 \endinput
```

5.7 VERSION HISTORY

```
29 v0.1 2012/11/24f. started
30     2012/11/26   code ready
31     2012/11/27   documented,
32                   \DoWithBaseExts -> \DoWithBasesExts
33
```