Addresslists with LaTeX

Tobias Spribille

May 14, 2008

1 User's Guide

`\ForEachAdress` The package `adrlist` provides the macro `ForEachAddress` as follows:

`\ForEachAdress{`⟨*addressfile*⟩`}{`⟨*commands*⟩`}`

The parameters are:

addressfile Name of a file containing the addresses in the following structure:

`<title>`	Title, degree, e.g. "Dipl.-Inf."
`<opening>`	how to address the person
`<sex>`	the sex, for grammatical correctness
`<first name>`	First name
`<name>`	Name
`-- begin address --`	delimiter
n lines of address	address with variable length
`-- end address --`	delimiter, to be typed exactly this way
Telephone: `<telephone number>`	These entries may appear in any order. They are distinguished through the keywords before the colon ":".
Telefax: `<telefax number>`	
EMail: `<eMail address>`	
Private number: `<private number>`	

commands LaTeX-commands to typeset the read data. You can use these commands defined by the package in order to access the address data:

`\Title`	Title
`\Opening`	Opening
`\Sex`	Sex
`\Firstname`	First name
`\Name`	Name
`\Address`	Address. The lines are separated by `\\`
`\Telephone`	Telephone number
`\Telefax`	Telefax number
`\EMail`	EMail address
`\PrivateNumber`	Private telephone number

2 The Macros

2.1 Declarations

This file provides the macro package `adrlist`. For various loops and conditions we require the `ifthen`-package.

```
\ProvidesPackage{adrlist}[1995/12/21]
\RequirePackage{ifthen}
```

Declare some variables: `Rest` controls the outer loop, which reads several entries out of the address database. If there are lines left in the file, it is true. `AddressLoop` is used for reading the adresses with variable length. `Communication` is a counter to repeat four times the recognition of the keywords for phone number, email etc.

```
\newboolean{AddressLoop}
\newboolean{Rest}\setboolean{Rest}{true}
\newcounter{Communication}
```

2.2 Helper macros

`\ifstringcompare` Compare two entire strings, given in the two parameter variables, and not only tokens, like TEX does.

```
\newcommand{\ifstringcompare}[4]{%
  \begingroup
    \let\protect=\noexpand
    \edef\0{#1}\edef\1{#2}%
  \expandafter\endgroup\ifx\0\1#3\else#4\fi}
```

`\concat` Concatenate two lines of the address, with `\\` as a delimiter, as it is required by e.g. the `letter` style.

```
\newcommand{\concat}[2]{#1\\#2}
```

`\keyword` Extract the keyword out of a line of the form ⟨*keyword*⟩`:`⟨*contents*⟩, i.e. the part before the colon. The pipe symbol may be replaced by any other character that won't appear in the address data.

```
\def\keyword#1:#2|{#1}
```

`\contents` Like `\keyword`. Extracts the contents of the line, e.g. the phone number.

```
\def\contents#1:#2|{#2}
```

2.3 The user macro

\ForEachAddress This macro is the one the user directly calls. It opens the file given in #1 and reads all the addresses in a loop. For each address the LaTeX-commands in #2 are executed to typeset the address in the document. After the whole file is read, it is closed again.

```
\newcommand{\ForEachAddress}[2]
{
 \newread\File \openin\File=#1
 \whiledo{\boolean{Rest}}
 { \ReadNextAddress{\File}
   \ifthenelse{\boolean{Rest}}{#2}{}
 }
 \closein\File
}
```

2.4 Main macrocode

\ReadNextAddress Here we read the next address out of the file given as #1 and save the read data into the respective variables. For the structure of the address file and the variable names, confer to section 1.

```
\newcommand{\ReadNextAddress}[1]{
 \ifthenelse{\boolean{Rest}}{
   \read#1 to \Title
   \ifeof#1\setboolean{Rest}{false}
   \else % If last address already is read, EOF should appear here
     \ifthenelse{\boolean{Rest}}
                {\read#1 to \Opening
                 \ifeof#1\setboolean{Rest}{false}\fi
                }{}
     \ifthenelse{\boolean{Rest}}
                {\read#1 to \Sex
                 \ifeof#1\setboolean{Rest}{false}\fi
                }{}
     \ifthenelse{\boolean{Rest}}
                {\read#1 to \Firstname
                 \ifeof#1\setboolean{Rest}{false}\fi
                }{}
     \ifthenelse{\boolean{Rest}}
                {\read#1 to \Name
                 \ifeof#1\setboolean{Rest}{false}\fi
                }{}
     \ifthenelse{\boolean{Rest}}
                {\read#1 to \Dummy
                 \ifeof#1\setboolean{Rest}{false}\fi
                }{}
     \setboolean{AddressLoop}{true}
     \def\Emptystring{} \let\Address=\Emptystring
     \def\Delimiter{-- end address -- }
```

Read Address, until `-- end address --` appears.

```
\whiledo{\boolean{AddressLoop}}
{
  \ifRest \read#1 to \Buffer \fi
  \ifeof#1\setboolean{Rest}{false}\fi
  \ifstringcompare{\Buffer}{\Delimiter}
                  {\setboolean{AddressLoop}{false}}{}
  \ifthenelse{\boolean{AddressLoop}}
  {
    \ifstringcompare{\Emptystring}{\Address}
    {
      \let\Address=\Buffer % Don't add delimiters the first time
    }
    {
       \begingroup     % keep redefinition of \\ local
       \let\protect=\noexpand
       \def\\{\noexpand\\}% make \\ not expandable
      \edef\x{\endgroup
       \def\noexpand\Address{\concat\Address\Buffer}%
      }\x
    }
  }{}
}
```

Four informations are left. The keyword is extracted from **\Buffer** and compared with the defined keywords. If the comparison succeeds, the correct variable is set.

```
\def\KeyWTelephone{Telephone}
\def\KeyWTelefax{Telefax}
\def\KeyWEMail{EMail}
\setcounter{Communication}{0}
\whiledo{\value{Communication}<4}%
{
  \ifthenelse{\boolean{Rest}}{
    \read#1 to \Buffer
    \edef\KeyW{\expandafter\keyword\Buffer|}
    \ifstringcompare{\KeyW}{\KeyWTelephone}
    {
      \edef\Telephone{\expandafter\contents\Buffer|}
    }
    {
      \ifstringcompare{\KeyW}{\KeyWTelefax}
      {
        \edef\Telefax{\expandafter\contents\Buffer|}
      }
      {
        \ifstringcompare{\KeyW}{\KeyWEMail}
        {
          \edef\EMail{\expandafter\contents\Buffer|}
        }
        {
```

```
97              \edef\PrivateNumber{\expandafter\contents\Buffer|}
98            }
99          }
100       }
101       \ifeof#1\setboolean{Rest}{false}\fi
102     }{}
103     \stepcounter{Communication}
104   }
105   \fi
106 }{}
107 }
```