The includex LaTeX package
extra ways to include files

Matt Swift <swift@alum.mit.edu>

Version: 0.6a Date: 1999/02/23
Documentation revision: 1999/02/23

Abstract

Includex allows you to include just the contents of one source file into another, ignoring the text outside the `\begin{document}` . . . `\end{document}` in the included file.

This package is useful but may have problems and is unsupported.

See also the *newclude* package.

Contents

I Discussion **2**

1 Including Files **2**

II Implementation **3**

2 Version control **3**

3 Requirements **3**

4 Disabling things **4**

5 New things **4**

Part I
Discussion

1 Including Files

\include* `\include` was always pretty useless to me because it did a `\clearpage` before and after the `\input`. The macro `\include*` is defined below to be like `\include` but without the `\clearpage` s.

The other thing we do is define a command `\includedoc`, which includes a file which is allowed to have its own `\documentclass` and `\begin{document}` and `\end{document}` and `\usepackage` commands.

This opens up a whole new world of truly modular LaTeX files. There is even nothing special about extracting the `document` environment, we can extract *any* environment, or perhaps even other things.

There are some problems, however. Maybe the `\usepackage` is necessary, maybe it would conflict. Same with other commands like the `\newcommand` family. It is simply not (yet) an issue in LaTeX design to consider the implications of parent files. The problems are not insurmountable. We can begin by working with a convention, and ultimately maybe make some changes in certain macros. I will be getting to all this. I admit I have not looked carefully to see whether the design specifications of LaTeX3 address these important issues.

The macros here will be useful in many circumstances, when the included files are themselves simple.

\includedoc
\includedoc*
\includedocskip
\includedocskip* There are now two main commands, `\includedoc` and `\includedocskip`, each with a `*` version analogous to the new `include*`. Each take a filename as an argument just like `\include`.

`\includedoc{`⟨*filename*⟩`}` reads everything in the file except for `\documentclass`, `\usepackage`, `\begin{document}`, `\end{document}` (and whatever follows it).

`\includedocskip {`⟨*filename*⟩`}` does the same except it also skips everything between `\documentclass` and `\begin{document}`.

\disable At present there is a makeshift way to ignore additional commands when using `\includedoc` and `\includedocskip`. If you want to cause `\foo` which takes no arguments to be disabled, say `\disable{\let\foo\relax}`. If `\foo` takes one mandatory argument, say `\let\foo\Gobble` instead. If `\foo` takes one optional and one mandatory, say `\let\foo\GobbleOM`. If `\foo` takes two mandatory arguments, say `\let\foo\GobbleMM`. For other permutations of arguments, you can extrapolate from the macros defined below.

Part II
Implementation

2 Version control

\fileinfo
\DoXUsepackagE
\HaveECitationS
\fileversion
\filedate
\docdate
\PPOptArg

These definitions must be the first ones in the file.

```
\def\fileinfo{extra ways to include files}
\def\DoXPackageS {}
\def\fileversion{v0.6a}
\def\filedate{1999/02/23}
\def\docdate{1999/02/23}
\edef\PPOptArg {%
  \filedate\space \fileversion\space \fileinfo
}
```

If we're loading this file from a `\ProcessDTXFile` command (see the *compsci* package), then `\JusTLoaDInformatioN` will be defined; othewise we assume it is not (that's why the FunkY NamE).

If we're loading from `\ProcessDTXFile`, we want to load the packages listed in `\DoXPackageS` (needed to typeset the documentation for this file) and then bail out. Otherwise, we're using this file in a normal way as a package, so do nothing. `\DoXPackageS`, if there are any, are declared in the `dtx` file, and, if you're reading the typeset documentation of this package, would appear just above. (It's OK to call `\usepackage` with an empty argument or `\relax`, by the way.)

```
\makeatletter% A special comment to help create bst files.  Don't change!
\@ifundefined{JusTLoaDInformatioN} {%
  }{% ELSE (we know the compsci package is already loaded, too)
  \UndefineCS\JusTLoaDInformatioN
  \SaveDoXVarS
  \eExpand\csname DoXPackageS\endcsname\In {%use \csname in case it's undefined
    \usepackage{#1}%
  }%
  \RestoreDoXVarS
  \makeatother
  \endinput
}% A special comment to help create bst files.  Don't change!
```

Now we check for LaTeX2e and declare the LaTeX package.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{includex}[\PPOptArg]
```

3 Requirements

Now we declare the package, check for LaTeX2e, and load the other packages needed.

```
\NeedsTeXFormat{LaTeX2e}
```

```
\ProvidesPackage{includex}
\RequirePackage{moredefs}
```

4 Disabling things

\disable This allows the disabling hacks.

```
\ReserveCS\DisableMonster  % This line is elsewhere for main package
\newcommand\disable[1] {%
  \g@addto@macro\DisableMonster{#1}%
}
```

5 New things

\sc@saved@include This is a simple redefinition of \include without the \clearpage commands.
\include Nothing tricky going on.
\sc@include
\sc@@include

```
\newlet\sc@saved@include\include
\renewcommand\include {%
  \@ifstar
      \sc@include
    \sc@saved@include
}
\newcommand\sc@include {%
  \relax
  \ifnum\@auxout=\@partaux
    \@latex@error{\string\include\space cannot be nested}\@eha
  \else
    \expandafter\sc@@include
  \fi
}
\newcommand\sc@@include[1] {%
  \if@filesw
    \immediate\write\@mainaux{\string\@input{#1.aux}}%
  \fi
  \@tempswatrue
  \if@partsw
    \@tempswafalse
    \edef\@tempb{#1}%
    \@for \@tempa:=\@partlist
      \do{\ifx\@tempa
            \@tempb
            \@tempswatrue
          \fi}%
  \fi
  \if@tempswa
    \let\@auxout\@partaux
    \if@filesw
      \immediate\openout\@partaux #1.aux%
```

```
      \immediate\write\@partaux{\relax}%
    \fi
    \@input@{#1.tex}%
    \@writeckpt{#1}%
    \if@filesw
% NOTE: Robin took out the \immediate here.
%      \immediate\closeout\@partaux
      \closeout\@partaux
    \fi
  \else
    \@nameuse{cp@#1}%
  \fi
  \let\@auxout\@mainaux
}
```

Now we start dealing with the much more tricky extraction of the `document` environment.

We start with considering how to quit inputting a file. The idea is to make the `\end{document}` command of the included file call `\endinput`. But there is a hitch that characters on the line after the `\end{document}` get inserted when you don't want them to. To beat that limitation, we have to do some work.

`\sc@radical@shutdown` We will add a bunch of commands to this macro, with the idea of `\catcode` ing everything and its brother to a comment. This would be a brute force method!

```
\ReserveCS\sc@radical@shutdown
```

First log a message that we're about to do some crazy things. In case something ever goes wrong, this might help.

```
\addto@macro\sc@radical@shutdown {%
  \PackageInfo{includex}{\protect\sc@radical@shutdown\space beginning}%
}
```

Now we start adding `\catcode` commands. These first two should be redundant; but just in case someone changed things. . . .

```
\addto@macro\sc@radical@shutdown{\catcode`\%=14}     % 14 = comment
\addto@macro\sc@radical@shutdown{\catcode`\^=7}      % 7 = superscript
```

`\sc@disable@char` Next, we define a command we weill use in a loop in a moment.

```
\newcommand\sc@disable@char[1] {%
  \addto@macro\sc@radical@shutdown
    {\catcode`#1=14}}  % 14 = comment
```

The following list contains every keyboard char except these three, which are treated specially: `%#`. The first is already a comment, and we handle the second in a moment. Each character in the following list is `\catcode` d to a comment:

```
\@tfor\sc@t@a:=abcdefghijklmnopqrstuvwxyz%
                ABCDEFGHIJKLMNOPQRSTUVWXYZ%
                ~!@$&*()_+-=[]|/?.,<>%
                1234567890%
```

```
                ‘’";:%
                \^\\\{\}\ % this is really the chars "^\{}" and space
    \do {\expandafter\sc@disable@char\sc@t@a}
```

We add # separately, because it's tricky or impossible to put it into the list we just used.

```
\sc@disable@char\#
```

We end the macro with \endinput. This has to come after all the previous, otherwise, TEX goes ahead and reads to the end of the line immediately, with regular catcodes. This is a good theory, I'm not sure it's necessary.

```
\addto@macro\sc@radical@shutdown{\endinput}
```

\sc@radical@shutdown@aftergroup We need to use \sc@radical@shutdown this way.

```
\newcommand\sc@radical@shutdown@aftergroup {%
  \aftergroup\sc@radical@shutdown
}
```

\sc@begindoc@toks
\sc@enddoc@toks These are for \AtBeginIncludedDocument and \AtEndIncludedDocument.

```
\newtoks\sc@begindoc@toks
\sc@begindoc@toks{}
\newtoks\sc@enddoc@toks
\sc@enddoc@toks{}
```

\AtBeginIncludedDocument The analogue of \AtBeginDocument.

```
\newcommand\AtBeginIncludedDocument[1] {%
    \sc@begindoc@toks\expandafter{\the\sc@begindoc@toks#1}%
}
```

\AtEndIncludedDocument The analogue of \AtEndDocument

```
\newcommand\AtEndIncludedDocument[1] {%
    \sc@enddoc@toks\expandafter{\the\sc@enddoc@toks#1}%
}
```

\includedoc
\includedoc* Now we start from the other end, defining the user command.

```
\newcommand\includedoc {%
% Robin's new:
  \@ifstar {%
      \def\sc@includebit@cmd{\include*}\sc@includebit
    }{% ELSE
    \let\sc@includebit@cmd\include\sc@includebit
  }%
}

% My old:
%      {\sc@includebit}
%    {\clearpage
%    \sc@includebit[\clearpage]}}
```

`\sc@includebit` The commented-out code is for some eventual extensions.

Interface is `sc@includebit[`⟨*postlude*⟩`]{`⟨*file*⟩`}` (where `[`⟨*postlude*⟩`]` is command(s) to be executed after the inclusion is complete); it is assumed that `sc@includebit@cmd` is either `\include` or `include*`

```
\newcounter{sc@count}
\newcounter{bit}
\newcommand\sc@includebit [2] [] {%
  \setcounter{sc@count}{\thebit}%
  \begingroup
    \DisableMonster
    \let\usepackage\GobbleOM
% Robin's new:
    \def\enddocument {%
      \makeatletter
      \the\sc@enddoc@toks
      \sc@radical@shutdown@aftergroup
    }
% My old:
% \let\enddocument\sc@radical@shutdown@aftergroup
    \let\documentclass\GobbleOM
    \let\documentstyle\GobbleOM  % why worry about branching if 2.09..
    \def\document {%
      \the\sc@begindoc@toks
    }%
    \def\IncludedJobname {#2}%
    \sc@includebit@cmd{#2}%
  \endgroup
  \par
  #1
}
```

`\includedocskip` `\includedocskip*` Now we make a similar command that *entirely* skips the preamble, whereas `\includedoc` only skipped those commands that we specifically disabled. Obviously this whole business of the preamble needs to be radically reconceived if convenient modularity of documents is to be had. At the moment, no distinction is made between things like `\newcommand` and `\usepackage` that should be exported to a parent file, and those things that should not.

```
\newcommand\includedocskip {%
  \@ifstar {%
      \sc@includebitskip
    }{% ELSE
    \clearpage
    \sc@includebitskip[\clearpage]%
  }%
}
```

I tried combining `\sc@includebitskip` and `\sc@includebit` but something changed when I used a conditional to do this. I'm sure the problem could be figured out and the two combined.

`\sc@includebitskip` We begin skipping with the `\documentclass` or `\documentstyle` command, and stop with the `\begin{document}` command. With some hacking to the new `verbatim` environment, this could be done. I haven't done this yet, so this is implemented by stuffing everything into a macro argument. I can't think of a `\begin` occurring between `\documentclass` the `\begin` document, but that will screw things up if it does.

The commented lines here are once again for a future extension.

```
\newcommand\sc@includebitskip [2] [] {%
  \setcounter{sc@count}{\thebit}%
  \begingroup
    \DisableMonster
    \let\enddocument\sc@radical@shutdown@aftergroup
    \long\def\documentclass ##1\begin{%
      \begingroup
      \def\@currenvir{document}%
      \GobbleM
    }% "{document}" is still to be read.
    \let\documentstyle\documentclass
    \include*{#2}%
  \endgroup
  \par
  #1\relax
}
```

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols

\# 92
\% 80
\@auxout 38, 59, 74
\@currenvir 161
\@eha 39
\@for 52
\@ifstar . . 32, 109, 147
\@ifundefined 10
\@input 46
\@input@ 64
\@latex@error 39
\@mainaux 46, 74
\@nameuse 72
\@partaux 38, 59, 61, 62, 68, 69
\@partlist 52
\@tempa 52, 53
\@tempb 51, 54
\@tempswafalse 50
\@tempswatrue . . . 48, 55
\@tfor 85
\@writeckpt 65
\\ 90
\{ 90
\} 90
\^ 81, 90

\␣ 90

A

\addto@macro 77, 80, 81, 83, 93
\aftergroup 95
\AtBeginIncludedDocument 101
\AtEndIncludedDocument 104

B

\begin 159
\begingroup 124, 156, 160

C

\catcode 80, 81, 84
\clearpage 118, 119, 150, 151
\closeout 68, 69
\csname 14

D

\def . . 1–5, 110, 128, 137, 140, 159, 161
\disable *2*, 26
\DisableMonster 26, 28, 125, 157
\do 53, 91
\docdate 1
\document 137
\documentclass 135, 159, 164
\documentstyle 136, 164
\DoXPackageS 2
\DoXUsepackagE 1

E

\edef 6, 51
\eExpand 14
\else 40, 71
\endcsname 14
\enddocument 128, 134, 158
\endgroup 142, 166
\endinput 19, 93
\expandafter 41, 91, 102, 105

F

\fi 42, 47, 56, 57, 63, 70, 73
\filedate 1
\fileinfo 1
\fileversion 1

G

\g@addto@macro 28
\GobbleM 162
\GobbleOM . 126, 135, 136

H

\HaveECitationS 1

I

\if@filesw . . 45, 60, 66
\if@partsw 49
\if@tempswa 58
\ifnum 38
\ifx 53
\immediate 46, 61, 62, 67, 68
\In 14
\include 30, 110, 112, 165
\include* *2*
\IncludedJobname . . 140
\includedoc *2*, 107
\includedoc* *2*, 107
\includedocskip . *2*, 146
\includedocskip* *2*, 146

J

\JusTLoaDInformatioN 12

L

\let . 59, 74, 112, 126, 134–136, 158, 164
\long 159

M

\makeatletter . . . 9, 129
\makeatother 18

N

\NeedsTeXFormat . 21, 23
\newcommand 27, 36, 44, 82, 94, 101, 104, 107, 122, 146, 154
\newcounter . . . 120, 121
\newlet 30
\newtoks 97, 99

O

\openout 61

P

\PackageInfo 78
\par 143, 167
\PPOptArg 1, 22
\protect 78
\ProvidesPackage 22, 24

R

\relax 37, 62, 168
\renewcommand 31
\RequirePackage . . . 25
\ReserveCS 26, 76
\RestoreDoXVarS . . . 17

S

\SaveDoXVarS 13
\sc@@include 30
\sc@begindoc@toks 97, 102, 138
\sc@disable@char 82, 91, 92
\sc@enddoc@toks 97, 105, 130
\sc@include 30
\sc@includebit 110, 112, 117, 119, 120
\sc@includebit@cmd 110, 112, 141
\sc@includebitskip 148, 151, 154
\sc@radical@shutdown 76, 95
\sc@radical@shutdown@afterwrite 94, 131, 134, 158
\sc@saved@include . 30
\sc@t@a 85, 91
\setcounter . . . 123, 155
\space 7, 39, 78
\string 39, 46

T

\the . . 102, 105, 130, 138
\thebit 123, 155

U

\UndefineCS 12
\usepackage 15, 126

W

\write 46, 62