

The `askinclude` package

Pablo A. Straub

Heiko Oberdiek*

2019-12-10 v2.6

Abstract

This package replaces `\includeonly` by an interactive user interface.

Contents

1 Documentation	2
1.1 User interface	2
1.2 Single asking mode	4
1.3 Regular expressions	4
1.3.1 Method <code>makematch</code>	4
1.3.2 Method <code>pdfmatch</code>	5
1.4 History	6
1.5 New features of version 2	6
1.6 Compatibility to version 1	7
2 Implementation	7
2.1 Header and catcodes	7
2.2 Delay action after preamble	8
2.3 Write entry in <code>.aux</code> file	8
2.4 Remember <code>\include</code> files	8
2.5 Redefine <code>\include</code>	10
2.6 Store answer and match method in <code>.aux</code> file	10
2.7 Support for <code>\@includeonly</code>	10
2.8 Stop if <code>\includeonly</code> is active	10
2.9 <code>\includeonly</code>	11
2.10 Options	11
2.11 Strings	12
2.12 Set defaults	12
2.13 Questions	13
2.13.1 Generic question header	13
2.13.2 Display files	14
2.13.3 Previous answer	14
2.13.4 Current regular expression support	15
2.13.5 Main question text	15
2.13.6 Main question	15
2.14 Evaluate answer of main question	16
2.15 Ask for each file	19

*Please report any issues at <https://github.com/ho-tex/askinclude/issues>

3	Installation	21
3.1	Download	21
3.2	Bundle installation	22
3.3	Package installation	22
3.4	Refresh file name databases	22
3.5	Some details for the interested	22
4	Acknowledgement	23
5	References	23
6	History	23
[1991/11/05 v0.95]	23
[1991/11/05 v0.99]	23
[1991/11/07 v1.0]	23
[1991/12/13 v1.1]	23
[1994/11/02 v1.2e]	24
[2007/10/15 v1.0 (askinc)]	24
[2007/10/23 v2.0]	24
[2008/08/11 v2.1]	24
[2011/12/02 v2.2]	24
[2016/05/14 v2.3]	24
[2018/11/01 v2.4]	24
[2019/12/09 v2.5]	24
[2019-12-10 v2.6]	24
7	Index	25

1 Documentation

1.1 User interface

The `\include` feature of L^AT_EX allows the division of the text body in separate compilation units. It is page based and stores counter settings in additional auxiliary files. `\includeonly` limits the actually included files to the ones, listed in its argument.

Example:

```
\documentclass{report}

\usepackage{askinlude}

% Don't use \include in the preamble.
% Instead \input can be used or make a package

\begin{document}
\include{title}
\include{introduction}
\include{summary}
\include{index}
\end{document}
```

Package `askinlude` adds an interactive user interface. The user is asked which files he want to include and sets `\includeonly` accordingly.

The first time the package is used the question looks like:

```
*****
*** Package askinlude Question ***
*****  

Previous answer (noregexp):  

[*]  

Regular expressions:  

--> [noregexp] disabled  

[makematch] enabled, using package 'makematch'  

[pdfmatch] enabled, using pdfTeX's \pdfmatch  

Which files do you want to include?  

[foo,bar] comma separated file or pattern list  

[*] all files  

[-] no files  

[?] ask for each file  

[] use the previous answer  

\answer=chapter2,summary
```

Square brackets are used as quotation symbols, they are not part of the answer. The answer by the user follows in the last line after `\answer=`.

[foo,bar]: A comma separated file list (also one file only) is passed to macro `\includeonly` directly. If regular expression support is enabled, the entries in the list are interpreted as file name patterns that will be matched against the file names in `\include`, see [section 1.3](#).

[*]: This is a shortcut for specifying *all* files.

[-]: This means *none* of the files is included.

[?]: The switches in single asking mode. The user will be asked at each `\include`. See section [section 1.2](#).

[]: Just pressing return (empty string) uses the default answer, that was stored in the auxiliary file (`.aux`) of the previous L^AT_EXrun with package `askinlude`. Without previous runs the default is [*] for all files.

[noregexp]: Turns regular expression support off.

[makematch]: Enables regular expression support package `makematch` provides.

[pdfmatch]: It is shown, if pdfTeX 1.30 or later is used. It relies on an undocumented feature `\pdfmatch` that may change in future versions of pdfTeX.

Package `askinlude` remembers the files of `\include` in the auxiliary file and the given answer. they are presented at the next run at the start of the question:

```
*****
*** Package askinlude Question ***
*****  

Files, found by previous run in \include:  

(-) title  

(-) introduction  

(-) chapter1  

(+) chapter2  

(-) chapter3
```

```
(+) summary
```

```
Previous answer (noregexp):  
[chapter2,summary]
```

```
...
```

(+) and (-) marks files that were and were not included in the previous run. The file list and marks are not always correct. For example, if \include statements are added or removed. If the method of regular expression support is different, then the marks only reflects the mode of the previous answer.

1.2 Single asking mode

The special answer ‘?’ trigger single asking mode, at the next \include the user is asked again:

```
*****  
*** Package askinclude Question ***  
*****  
  
Include 'title'? [y]es, [n]o, [A]ll, [N]one, [D]efault (n):  
  
\answer=
```

Use the letter inside the square brackets as answer. An empty string uses the default answer, the letter in parentheses. The answers are:

[y]: The file is included.

[n]: The file is excluded.

[A]: This file and all following files are included.

[N]: This file and all following files are excluded.

[D]: Use the default answer for this and all following files.

If the file is known from the previous run, then the default is ‘y’ or ‘n’ corresponding to the mark ‘(+)’ or ‘(-)’ in the list of previous files. Otherwise ‘y’ is used as default.

1.3 Regular expressions

Three methods are available that can also set by package options:

noregexp: Regular expression support is disabled.

makematch: Use package makematch.

pdfmatch: Use pdftEX’s \pdfmatch if available.

Without options the value remembered in the auxiliary file is used, otherwise the default is noregexp.

1.3.1 Method **makematch**

Package **makematch** (part of **qstest**), written by David Kastrup provides pattern matching based on wildcards:

[*]: The star matches zero or more characters. Thus a lonely star matches anything.

[!]: The exclamation mark at the begin of a pattern negates it, thus the matching files are excluded.

The pattern list is evaluated for a file name from left to right. Always a file name must match the whole pattern. The file is included if it is matched by a pattern and not excluded afterwards. Examples:

Pattern list	File names	Result
foobar	foobar ,dummy	foobar
foo*	foobar , xfoobar	foobar
foo	xfoobar	xfoobar
f*b*r	foobar	foobar
*	any	any
!foobar	foobar , other	other
!foo*	xfoobar	xfoobar
in*,!*dex	intro , index	intro
!index,in*	intro , index	intro , index
*,!chap2	chap1 , chap2 , chap3	chap1 , chap3

The example in the last line shows, how the behaviour of package **excludeonly** [2] can be simulated.

The documentation of package **makematch** tells the the empty pattern does not match anything. But if you answer the question with an empty string the answer of the previous run is used. Also curly braces are not reliable because the can get stripped during the processing of the answer. Using a pair of empty curly braces or the comma for two empty patterns currently works, but this is specific for the current implementation and may change in future. But the special answer ‘-’ is available.

1.3.2 Method **pdfmatch**

In pdfTeX 1.30 the new primitive **\pdfmatch** was introduced. It is not documented yet, because it was not decided if it remains and which implementation of regular expressions should be used. Thus the situation may change in later versions of pdfTeX. Currently (1.30 until 1.40) it uses a library for POSIX 1003.2 regular expressions, see manual page “**regex(7)**”.

If a pattern in the pattern list is invalid, the package prints a warning and removes the pattern from the list. Actually there are two warnings, the other warning is generated by pdfTeX itself.

In opposite to **makematch** the pattern only needs to match the specified part of the file name. For matching the whole name, the symbols ‘^’ and ‘\$’ for begin and end of string must be included.

Additionally an initial exclamation mark ‘!’ implements the same logic as method **makematch**. The match result is inverted, the matched file excluded.

Examples:

Pattern list	File names	Result
foobar	foobar,dummy	foobar
foo.*	foobar,xfoobar	foobar,xfoobar
^foo.*	foobar,xfoobar	foobar
foo\$	foo,foobar	foo
fo?bar	fbar,fobar,foobar	fbar,fobar
fo+bar	fbar,fobar,foobar	fobar,foobar
fo*bar	fbar,fobar,foobar	fbar,fobar,foobar
chap3	chap3,chap32	chap1,chap3,chap32
chap[13]	chap1,chap2,chap3	chap1,chap3
chap(1 3)\$	chap1,chap3,chap33	chap1,chap3
.*	any	any
!foobar	foobar,other	other
in.*,!dex	intro,index	intro
!index,in.*	intro,index	intro,index
.*,!chap2	chap1,chap2,chap3	chap1,chap3
.*,!bla	foo,bla,blabla	foo
.*,!^bla\$	foo,bla,blabla	foo,blabla

The example in the last line shows, how the behaviour of package `excludeonly` [2] can be simulated. As the third last line shows, that sometimes ‘^’ and ‘\$’ can be omitted, but this will not work always, shown in the second last line.

1.4 History

Package `askinlude` was invented by Pablo A. Straub in 1991 and maintained until 1994.

In October 2007 it was removed from “TeX Live” because of the no-commercial license [1]. Herbert Voß initiated a search for the author in the newsgroup `comp.text.tex` [3] to ask him to change the license.

It seems that the author could not be reached any more. Joseph Wright suggested a reimplementation that I (HO) have done as package `askinc`. Soon it was enhanced by feature requests.

Then Ulrike Fischer found email addresses of the original author and Jim Hefferon succeeded in contacting him. He allowed a change of the license to LPPL and the transfer of the maintainership to someone else.

Thus I merged both packages as `askinlude` in a new major version 2. The license is LPPL.

1.5 New features of version 2

- License is now LPPL.
- A list of `\include` files that are detected by the previous run is printed before the question.
- If the special answer ‘?’ is given, then the user is asked at each `\include`.
- Support for regular expressions.
- The remaining stuff in the `.aux` file does not cause errors, if the package is removed.
- The use of `\includeonly` before the package loading time is detected and respected by taking precedence. After the package is loaded, `\includeonly` is disabled and prints a warning.

1.6 Compatibility to version 1

- The user interface of version 1 with its special answers ‘*’, ‘-’, and ‘’ remain unchanged, but it is extended by version 2 with the special answer ‘?’ and support for regular expressions can be enabled.
- As reimplementations the internal macros have changed of course.
- .aux files that are written by version 1.x will work with version 2.x because \@includeonly is recognized.
- L^AT_EX 2.09: The package works as \documentstyle option in compatibility mode of L^AT_EX 2_E. It is possible that it will not work in pure L^AT_EX 2.09.

2 Implementation

2.1 Header and catcodes

```
1 (*package)
2 \NeedsTeXFormat{LaTeX2e}
3 \begingroup\catcode61\catcode48\catcode32=10\relax%
4 \catcode13=5 % ^~M
5 \endlinechar=13 %
6 \catcode123=1 %
7 \catcode125=2 %
8 \catcode64=11 %
9 \def\x{\endgroup
10 \expandafter\edef\csname AskInc@AtEnd\endcsname{%
11   \endlinechar=\the\endlinechar\relax
12   \catcode13=\the\catcode13\relax
13   \catcode32=\the\catcode32\relax
14   \catcode35=\the\catcode35\relax
15   \catcode61=\the\catcode61\relax
16   \catcode64=\the\catcode64\relax
17   \catcode123=\the\catcode123\relax
18   \catcode125=\the\catcode125\relax
19 }%
20 }%
21 \x\catcode61\catcode48\catcode32=10\relax%
22 \catcode13=5 % ^~M
23 \endlinechar=13 %
24 \catcode35=6 % #
25 \catcode64=11 %
26 \catcode123=1 %
27 \catcode125=2 %
28 \def\TMP@EnsureCode#1#2{%
29   \edef\AskInc@AtEnd{%
30     \AskInc@AtEnd
31     \catcode#1=\the\catcode#1\relax
32   }%
33   \catcode#1=#2\relax
34 }
35 \TMP@EnsureCode{33}{12}!
36 \TMP@EnsureCode{39}{12} '
37 \TMP@EnsureCode{40}{12} (
38 \TMP@EnsureCode{41}{12} )
39 \TMP@EnsureCode{42}{12} *
40 \TMP@EnsureCode{43}{12} +
```

```

41 \TMP@EnsureCode{44}{12}%
42 \TMP@EnsureCode{45}{12}%
43 \TMP@EnsureCode{46}{12}%
44 \TMP@EnsureCode{47}{12}%
45 \TMP@EnsureCode{58}{12}%
46 \TMP@EnsureCode{62}{12}%
47 \TMP@EnsureCode{63}{12}%
48 \TMP@EnsureCode{91}{12}%
49 \TMP@EnsureCode{93}{12}%
50 \TMP@EnsureCode{96}{12}%
51 \edef\AskInc@AtEnd{\AskInc@AtEnd\noexpand\endinput}
52 \ProvidesPackage{askinlude}%
53 [2019-12-10 v2.6 Interactive asking of included files]%

    Luatex compat
54 \ifx\pdfextension\undefined\else
55     \protected\def\pdflastobj {\numexpr\pdffeedback lastobj\relax}
56     \protected\def\pdfnames {\pdfextension names }
57     \protected\def\pdfobj {\pdfextension obj }
58     \let\pdfoutput \outputmode
59 \fi

```

2.2 Delay action after preamble

`\AskInc@AtBeginDocument` Some actions such as `.aux` file writing cannot be performed in the preamble. `\AtBeginDocument` delays such action until `\begin{document}`. However `\AtBeginDocument` must not be used after that time (`\onlypreamble`). Thus `\AskInc@AtBeginDocument` uses `\AtBeginDocument` in the preamble and afterwards executes the code immediately.

```

60 \AtBeginDocument{%
61     \let\AskInc@AtBeginDocument\@firstofone
62 }
63 \providecommand*\AskInc@AtBeginDocument{\AtBeginDocument}

```

2.3 Write entry in `.aux` file

`\AskInc@WriteAux` The first argument is written to the main `.aux` file and the second argument is executed if writing was possible. Remember `\nofiles` prevents writing to auxiliary files.

```

64 \def\AskInc@WriteAux#1#2{%
65     \AskInc@AtBeginDocument{%
66         \if@filesw
67             \immediate\write\@mainaux{#1}%
68             #2%
69     \fi
70 }
71 }

```

2.4 Remember `\include` files

`\AskInc@CharHash`

```

72 \edef\AskInc@CharHash{\string#}

```

`\AskInc@WriteDefAddList`

```

73 \def\AskInc@WriteDefAddList{%
74     \AskInc@WriteAux{%

```

```

\AskInc@AddList
 75   \string\providecommand*\string\AskInc@AddList[2]{%
 76     \string\ifcase\string\ifx\AskInc@CharHash1\string@\undefined
 77       \string\else
 78         \string\ifx\AskInc@CharHash1\string\relax
 79       \string\else
 80         \string\ifx\AskInc@CharHash1\string@\empty
 81       \string\else
 82         1%
 83       \string\fi
 84     \string\fi
 85   \string\fi
 86   0 %
 87   \string\gdef\AskInc@CharHash1{\AskInc@CharHash2}%
 88   \string\else
 89     \string\g@addto@macro\AskInc@CharHash1{,\AskInc@CharHash2}%
 90   \string\fi
 91 }
 92 }{%
 93   \global\let\AskInc@WriteDefAddList\relax
 94 }
 95 }

\AskInc@WriteDefAddFile
 96 \def\AskInc@WriteDefAddFile{%
 97   \AskInc@WriteDefAddList
 98   \AskInc@WriteAux{%

\AskInc@AddFile
 99   \string\providecommand\string\AskInc@AddFile{%
100     \string\AskInc@AddList\string\AskInc@Files
101   }%
102 }{%
103   \global\let\AskInc@WriteDefAddFile\relax
104 }
105 }

\AskInc@AddFile
106 \AtBeginDocument{%
107   \global\let\AskInc@AddFile\@gobble
108 }

\AskInc@WriteAddFile
109 \def\AskInc@WriteAddFile#1{%
110   \AskInc@WriteDefAddFile
111   \AskInc@WriteAux{%
112     \string\AskInc@AddFile{#1}%
113   }{%
114 }

\AskInc@Files
115 \providecommand*\{\AskInc@Files}{}

```

Summary: Now `\AskInc@Files` is always defined and either empty or contains a comma separated list of `\include` files of previous run.

The `\include` files are remembered by using `\AskInc@WriteAddFile` at each `\include`.

2.5 Redefine \include

```
\AskInc@OrgInclude
116 \newcommand{\AskInc@OrgInclude}{}%
117 \let\AskInc@OrgInclude\include

\AskInc@AskFile
118 \let\AskInc@AskFile\@gobble

\AskInc@MatchFile
119 \let\AskInc@MatchFile\@gobble

\include
120 \renewcommand*\include}[1]{%
121   \AskInc@WriteAddFile{#1}%
122   \AskInc@AskFile{#1}%
123   \AskInc@MatchFile{#1}%
124   \AskInc@OrgInclude{#1}%
125 }
```

2.6 Store answer and match method in .aux file

```
126 \AtEndOfPackage{%
127   \AtBeginDocument{%
128     \@ifundefined{AskInc@Answer}{}{%
129       \AskInc@WriteAux{%
130         \string\gdef\string\AskInc@Answer{\AskInc@Answer}}%
131       }{}%
132     }%
133     \@ifundefined{AskInc@MatchMethod}{}{%
134       \AskInc@WriteAux{%
135         \string\gdef\string\AskInc@MatchMethod{\AskInc@MatchMethod}}%
136       }{}%
137     }%
138   }%
139 }
```

2.7 Support for \cincludeonly

\cincludeonly Macro \cincludeonly is defined for compatibility reasons only, if the .aux file was written by versions before 2 of package askinlude.

```
140 \providecommand*\cincludeonly}[1]{%
141   \gdef\AskInc@Answer{#1}%
142   \ifx\AskInc@Answer\empty
143     \gdef\AskInc@Answer{-}%
144   \fi
145   \global\let\AskInc@MatchMethod\empty
146 }
```

2.8 Stop if \includeonly is active

If \includeonly is used before, the questions are skipped and the data in the main auxiliary file is preserved.

```
147 \if@partsw
148   \PackageWarningNoLine{askinlude}{%
149     Previous \string\includeonly\space detected,\MessageBreak
150     therefore asking for files is suppressed%
151   }%
```

```

Dummy options.

152 \DeclareOption{makematch}{}%
153 \DeclareOption{pdfmatch}{}%
154 \DeclareOption{noregexp}{}%
155 \ProcessOptions*%


Preserve \AskInc@MatchFiles.

156 \@ifundefined{AskInc@MatchFiles}{%
157 }{%
158   \ifx\AskInc@MatchMethod\empty
159   \else
160     \AskInc@WriteAux{%
161       \string\gdef\string\AskInc@MatchFiles{\AskInc@MatchFiles}%
162     }{%
163     \fi
164   }%
165 \expandafter\AskInc@AtEnd
166 \fi%

```

2.9 \includeonly

```

\AskInc@OrgIncludeOnly

167 \newcommand{\AskInc@OrgIncludeOnly}{}%
168 \let\AskInc@OrgIncludeOnly\includeonly

\includeonly

169 \renewcommand*\includeonly[1]{%
170   \PackageWarning{askinlude}{%
171     string\includeonly space is managed by package 'askinlude',%
172     \MessageBreak
173     therefore it is ignored%
174   }%
175 }

```

2.10 Options

```

\AskInc@StringNoRegExp

176 \def\AskInc@StringNoRegExp{noregexp}

\AskInc@StringMakeMatch

177 \def\AskInc@StringMakeMatch{makematch}

\AskInc@StringPdfMatch

178 \def\AskInc@StringPdfMatch{pdfmatch}

\ifAskInc@PdfMatchAvailable

179 \newif\ifAskInc@PdfMatchAvailable
180 \begingroup\expandafter\expandafter\expandafter\endgroup
181 \expandafter\ifx\csname pdfmatch\endcsname\relax
182 \else
183   \AskInc@PdfMatchAvailabletrue
184 \fi

185 \DeclareOption{makematch}{%
186   \let\AskInc@MatchOption\AskInc@StringMakeMatch
187 }
188 \ifAskInc@PdfMatchAvailable

```

```

189  \DeclareOption{pdfmatch}{%
190    \let\AskInc@MatchOption\AskInc@StringPdfMatch
191  }%
192 \else
193  \DeclareOption{pdfmatch}{%
194    \PackageWarningNoLine{askininclude}{%
195      Option 'pdfmatch' is not available because\MessageBreak
196      your pdfTeX does not provide \string\pdfmatch
197    }%
198  }%
199 \fi
200 \DeclareOption{noregexp}{%
201  \let\AskInc@MatchOption\@empty
202 }
203 \ProcessOptions*

```

2.11 Strings

```
\AskInc@CharAsterisk
204 \def\AskInc@CharAsterisk{*}
```

```
\AskInc@CharMinus
205 \def\AskInc@CharMinus{-}
```

```
\AskInc@CharPlus
206 \def\AskInc@CharPlus{+}
```

```
\AskInc@CharQuestion
207 \def\AskInc@CharQuestion{?}
```

```
\AskInc@CharExclam
208 \def\AskInc@CharExclam{!}
```

```
\AskInc@CharYes
209 \def\AskInc@CharYes{y}
```

```
\AskInc@CharNo
210 \def\AskInc@CharNo{n}
```

```
\AskInc@CharAll
211 \def\AskInc@CharAll{A}
```

```
\AskInc@CharNone
212 \def\AskInc@CharNone{N}
```

```
\AskInc@CharDefault
213 \def\AskInc@CharDefault{D}
```

2.12 Set defaults

```
\AskInc@SanitizeAnswer
214 \def\AskInc@SanitizeAnswer#1#2{%
215   \begingroup
216     \escapechar='\\%
217     \edef\[{\string\[}%
218     \edef\][]{\string\]}%
```

```

219  \edef\({\string\()%
220  \edef\){\string\))%
221  \edef\*{\string\*}%
222  \edef\+{\string\+}%
223  \edef\?{\string\?}%
224  \edef\|{\string\|}%
225  \edef\^{\string\^}%
226  \edef\${\string\$}%
227  \edef\{\{\string\{}%
228  \edef\}\{\string\}}%
229  \edef\.\{\string\.\}%
230  \xdef#1{%
231      \expandafter\expandafter\expandafter\zap@space
232      \expandafter#2\space\empty
233  }%
234  \endgroup
235 }

236 \AtBeginDocument{%
237  \@ifundefined{AskInc@Answer}{%
238      \global\let\AskInc@Answer\AskInc@CharAsterisk
239  }{%
240      \AskInc@SanitizeAnswer\AskInc@Answer\AskInc@Answer
241  }%
242  \global\let\AskInc@PreviousAnswer\AskInc@Answer
243  \@ifundefined{AskInc@MatchMethod}{%
244      \global\let\AskInc@MatchMethod\empty
245  }{%
246  \global\let\AskInc@PreviousMatchMethod\AskInc@MatchMethod
247  \@ifundefined{AskInc@MatchOption}{%
248      \let\AskInc@MatchOption\AskInc@MatchMethod
249  }{%
250  \ifx\AskInc@MatchOption\AskInc@StringPdfMatch
251      \ifAskInc@PdfMatchAvailable
252      \else
253          \let\AskInc@MatchOption\empty
254      \fi
255  \fi
256  \@ifundefined{AskInc@MatchFiles}{%
257      \let\AskInc@MatchFiles\empty
258  }{%
259 }

```

2.13 Questions

2.13.1 Generic question header

\AskInc@QuestionHeader

```

260 \def\AskInc@QuestionHeader{%
261  \typeout{%
262  \typeout{*****}
263  \typeout{*** Package askinlude Question ***}
264  \typeout{*****}
265  \typeout{%
266 }

```

2.13.2 Display files

```
\AskInc@TypeoutFiles
267 \def\AskInc@TypeoutFiles{%
268   \global\let\AskInc@PreviousMatches\@empty
269   \ifx\AskInc@Files\@empty
270   \else
271     \typeout{Files, found by previous run in \string\include:}%
272     \@for\AskInc@File:=\AskInc@Files\do{%
273       \ifx\AskInc@Answer\AskInc@CharAsterisk
274         \let\AskInc@Mark\AskInc@CharPlus
275       \else
276         \let\AskInc@Mark\AskInc@CharMinus
277         \ifx\AskInc@Answer\AskInc@CharMinus
278         \else
279           \ifx\AskInc@MatchMethod\@empty
280             \expandafter\@firstoftwo
281           \else
282             \expandafter\@secondoftwo
283           \fi
284           {\@for\AskInc@Temp:=\AskInc@Answer}%
285           {\@for\AskInc@Temp:=\AskInc@MatchFiles}%
286           \do{%
287             \ifx\AskInc@File\AskInc@Temp
288               \let\AskInc@Mark\AskInc@CharPlus
289             \fi
290           }%
291           \fi
292           \fi
293           \typeout{ \cspaces(\AskInc@Mark) \AskInc@File}%
294           \ifx\AskInc@Mark\AskInc@CharPlus
295             \ifx\AskInc@PreviousMatches\@empty
296               \global\let\AskInc@PreviousMatches\AskInc@File
297             \else
298               \expandafter\g@addto@macro
299               \expandafter\AskInc@PreviousMatches{%
300                 \expandafter,\AskInc@File
301               }%
302             \fi
303           \fi
304         }%
305         \typeout{}%
306       \fi
307 }
```

2.13.3 Previous answer

```
\AskInc@TypeoutPreviousAnswer
308 \def\AskInc@TypeoutPreviousAnswer{%
309   \typeout{%
310     Previous answer (%
311     \ifx\AskInc@MatchMethod\@empty
312       noregexp%
313     \else
314       \AskInc@MatchMethod
315     \fi
316   ):%
317 }
```

```

318   \typeout{ \@spaces[\AskInc@Answer]}%
319   \typeout{}%
320 }

```

2.13.4 Current regular expression support

```
\AskInc@TypeoutMatchMethod
321 \def\AskInc@TypeoutMatchMethod{%
322   \typeout{Regular expressions:}%
323   \AskInc@TypeoutMethod\@empty\noregexp{%
324     \space disabled%
325   }%
326   \AskInc@TypeoutMethod\AskInc@StringMakeMatch{makematch}{%
327     enabled, using package ‘makematch’%
328   }%
329   \ifAskInc@PdfMatchAvailable
330     \AskInc@TypeoutMethod\AskInc@StringPdfMatch{pdfmatch}{%
331       \space enabled, using pdfTeX’s \string\pdfmatch
332     }%
333   \fi
334   \typeout{}%
335 }
```

```
\AskInc@TypeoutMethod
336 \def\AskInc@TypeoutMethod#1#2#3{%
337   \typeout{%
338     \ifx#1\AskInc@MatchOption \space--> \else\@spaces\space\fi
339     [#2] #3% hash-ok
340   }%
341 }
```

2.13.5 Main question text

```
\AskInc@TypeoutMainQuestion
342 \def\AskInc@TypeoutMainQuestion{%
343   \typeout{Which files do you want to include?}%
344   \typeout{ \@spaces[foo,bar] \space\space
345     comma separated file or pattern list}%
346   \typeout{ \@spaces[*] \@spaces\@spaces all files}%
347   \typeout{ \@spaces[-] \@spaces\@spaces no files}%
348   \typeout{ \@spaces[?] \@spaces\@spaces ask for each file}%
349 }
```

2.13.6 Main question

```
\AskInc@MainQuestion
350 \def\AskInc@MainQuestion{%
351   \AskInc@QuestionHeader
352   \AskInc@TypeoutFiles
353   \AskInc@TypeoutPreviousAnswer
354   \AskInc@TypeoutMatchMethod
355   \AskInc@TypeoutMainQuestion
356   \begingroup
357     \@makeother\*%
358     \@makeother\-%
359     \@makeother\?%
360     \@makeother\,%
```

```

361  \@makeother\!%
362  \@makeother\~%
363  \@makeother\$%
364  \@makeother\(%
365  \@makeother\)%%
366  \@makeother\+%
367  \@makeother\[%
368  \@makeother\]%
369  \@makeother\\%
370  \@makeother\|%
371  \@makeother\::%
372  \@makeother\.%%
373  \typein[\answer]%
374  { \cspaces[] \space\cspaces\cspaces use the previous answer}%
375  \typeout{}%
376  \AskInc@SanitizeAnswer\AskInc@@Answer\answer
377  \endgroup
378  \let\AskInc@Next\AskInc@MainQuestion
379  \ifx\AskInc@@Answer\AskInc@StringNoRegExp
380  \let\AskInc@MatchOption\empty
381  \else
382  \ifx\AskInc@@Answer\AskInc@StringMakeMatch
383  \let\AskInc@MatchOption\AskInc@StringMakeMatch
384  \else
385  \ifx\AskInc@@Answer\AskInc@StringPdfMatch
386  \ifAskInc@PdfMatchAvailable
387  \let\AskInc@MatchOption\AskInc@StringPdfMatch
388  \else
389  \PackageWarning{askinlude}{%
390  Method 'pdfmatch' is not available because\MessageBreak
391  your pdfTeX does not provide \string\pdfmatch
392  }%
393  \let\AskInc@MatchOption\empty
394  \fi
395  \else
396  \ifx\AskInc@@Answer\empty
397  \global\let\AskInc@@Answer\AskInc@Answer
398  \fi
399  \let\AskInc@Next\AskInc@MainAnswer
400  \fi
401  \fi
402  \fi
403  \AskInc@Next
404 }

```

2.14 Evaluate answer of main question

```

\AskInc@MainAnswer
405 \def\AskInc@MainAnswer{%
406  \ifx\AskInc@@Answer\AskInc@CharQuestion
407  \global\let\AskInc@Answer\empty
408  \global\let\AskInc@MatchMethod\empty
409  \global\let\AskInc@AskFile\AskInc@DoAskFile
410  \AskInc@OrgIncludeOnly{}%
411  \else
412  \global\let\AskInc@Answer\AskInc@@Answer
413  \ifx\AskInc@@Answer\AskInc@CharAsterisk
414  \global\let\AskInc@MatchMethod\empty

```

```

415   \else
416     \ifx\AskInc@Answer\AskInc@CharMinus
417       \global\let\AskInc@MatchMethod\@empty
418       \AskInc@OrgIncludeOnly{}%
419     \else
420       \global\let\AskInc@MatchMethod\AskInc@MatchOption
421       \ifx\AskInc@MatchOption\@empty
422         \expandafter\AskInc@OrgIncludeOnly\expandafter{%
423           \AskInc@Answer
424         }%
425       \else
426         \AskInc@MatchInit
427         \AskInc@OrgIncludeOnly{}%
428       \fi
429     \fi
430   \fi
431 \fi
432 \let\AskInc@Include\AskInc@DoInclude
433 }

\AskInc@MatchInit
434 \def\AskInc@MatchInit{%
435   \RequirePackage{kvsetkeys}[2007/09/29]%
436   \expandafter\comma@normalize\expandafter{\AskInc@Answer}%
437   \global\let\AskInc@MatchList\comma@list

\AskInc@MatchFile
438 \def\AskInc@MatchFile##1{%
439   \AskInc@DoMatch{##1}{%
440     \AskInc@WriteAddMatch{##1}%
441     \ifx\@partlist\@empty
442       \gdef\@partlist{##1}%
443     \else
444       \g@addto@macro\@partlist{,##1}%
445     \fi
446   }%
447 }%

448 \ifx\AskInc@MatchMethod\AskInc@StringMakeMatch
449   \AskInc@MakeMatchInit
450 \else
451   \AskInc@PdfMatchInit
452 \fi
453 \let\AskInc@MatchInit\@undefined
454 }

\AskInc@MakeMatchInit
455 \def\AskInc@MakeMatchInit{%
456   \RequirePackage{makematch}%

\AskInc@DoMatch
457 \def\AskInc@DoMatch##1{%
458   \def\AskInc@Temp{%
459     \MakeMatcher[,] \AskInc@Temp
460   }%
461   \expandafter\AskInc@Temp\expandafter{\AskInc@MatchList}%
462   \AskInc@Temp{##1}\@firstofone\@gobble
463 }%

```

```

464   \let\AskInc@MakeMatchInit@\undefined
465 }

\AskInc@PdfMatchInit
466 \def\AskInc@PdfMatchInit{%
  \ifAskInc@Found
    467   \newif\ifAskInc@Found

\AskInc@DoMatch
468   \def\AskInc@DoMatch##1{%
  469     \AskInc@Foundfalse
  470     \expandafter\comma@parse@normalized\expandafter
  471     {\AskInc@MatchList}{\AskInc@CheckMatch{##1}}%
  472     \ifAskInc@Found
  473       \expandafter\@firstofone
  474     \else
  475       \expandafter\@gobble
  476     \fi
  477   }%

\AskInc@CheckMatch
478   \def\AskInc@CheckMatch##1##2{%
  479     \AskInc@@CheckMatch##2\@empty\@nil{##1}%
  480   }%

\AskInc@@CheckMatch
481   \def\AskInc@@CheckMatch##1##2\@nil##3{%
  482     \def\AskInc@Temp{##1}%
  483     \ifx\AskInc@Temp\AskInc@CharExclam
  484       \ifcase\pdfmatch{##2}{##3}=1 %
  485         \or
  486           \AskInc@Foundfalse
  487         \else
  488           \AskInc@InvalidPattern{##2}{##1##2}%
  489         \fi
  490       \else
  491         \ifcase\pdfmatch{##1##2}{##3} %
  492           \or
  493             \AskInc@Foundtrue
  494           \else
  495             \AskInc@InvalidPattern{##1##2}{##1##2}%
  496           \fi
  497         \fi
  498   }%

\AskInc@InvalidPattern Prints a warning with the regular expression and removes it from the list to prevent
  further warnings for the same expression.
499   \def\AskInc@InvalidPattern##1##2{%
  500     \edef\AskInc@Temp{##1}%
  501     \PackageWarningNoLine{askinlude}{%
  502       Invalid pattern '\AskInc@Temp' for \string\pdfmatch
  503     }%
  504     \edef\AskInc@Temp{##2}%
  505     \global\expandafter\let\expandafter\AskInc@MatchList
  506     \expandafter\@empty
  507     \expandafter\comma@parse@normalized\expandafter
  508     {\AskInc@MatchList}\AskInc@FilterPattern
  509   }%

```

```

\AskInc@FilterPattern
510  \def\AskInc@FilterPattern##1{%
511    \edef\AskInc@Pattern{\#1}%
512    \ifx\AskInc@Temp\AskInc@Pattern
513    \else
514      \ifx\AskInc@MatchList\empty
515        \global\let\AskInc@MatchList\AskInc@Pattern
516      \else
517        \expandafter\g@addto@macro\expandafter\AskInc@MatchList{%
518          \expandafter,\AskInc@Pattern
519        }%
520      \fi
521    \fi
522  }%
523  \let\AskInc@PdfMatchInit\undefined
524 }

\AskInc@WriteDefAddMatch
525 \def\AskInc@WriteDefAddMatch{%
526   \AskInc@WriteDefAddList
527   \AskInc@WriteAux{%
528     \string\providacommand\string\AskInc@AddMatch{%
529       \string\AskInc@AddList\string\AskInc@MatchFiles
530     }%
531   }{%
532     \global\let\AskInc@WriteDefAddMatch\relax
533   }%
534 }

\AskInc@AddMatch
535 \AtBeginDocument{%
536   \global\let\AskInc@AddMatch\@gobble
537 }

\AskInc@WriteAddMatch
538 \def\AskInc@WriteAddMatch#1{%
539   \AskInc@WriteDefAddMatch
540   \AskInc@WriteAux{%
541     \string\AskInc@AddMatch{#1}%
542   }{}%
543 }

\AskInc@SaveAnswer
544 \def\AskInc@SaveAnswer{%
545   \AskInc@WriteAux{%
546     \string\gdef\string\AskInc@Answer{\AskInc@Answer}%
547   }{}%
548 }

```

2.15 Ask for each file

```

\AskInc@PreviousAnswer
549 \@ifundefined{AskInc@PreviousAnswer}{%
550   \let\AskInc@PreviousAnswer\AskInc@CharAsterisk
551 }{}%

```

```

\AskInc@PreviousMatchMethod
552 \@ifundefined{\AskInc@PreviousMatchMethod}{%
553   \let\AskInc@PreviousMatchMethod\empty
554 }{ }

\AskInc@DoAskFile
555 \def\AskInc@DoAskFile#1{%
556   \edef\AskInc@File{\#1}%
557   \ifx\AskInc@PreviousAnswer\AskInc@CharAsterisk
558     \let\AskInc@Default\AskInc@CharYes
559   \else
560     \let\AskInc@Default\AskInc@CharNo
561   \ifx\AskInc@PreviousAnswer\AskInc@CharMinus
562   \else
563     \ifx\AskInc@PreviousMatchMethod\empty
564       \expandafter\@firstoftwo
565     \else
566       \expandafter\@secondoftwo
567     \fi
568   {\@for\AskInc@Temp:=\AskInc@PreviousAnswer}%
569   {\@for\AskInc@Temp:=\AskInc@MatchFiles}%
570   \do{%
571     \ifx\AskInc@Temp\AskInc@File
572       \let\AskInc@Default\AskInc@CharYes
573     \fi
574   }%
575   \fi
576   \ifx\AskInc@Default\AskInc@CharNo
577     \let\AskInc@Default\AskInc@CharYes
578   \@for\AskInc@Temp:=\AskInc@Files\do{%
579     \ifx\AskInc@Temp\AskInc@File
580       \let\AskInc@Default\AskInc@CharNo
581     \fi
582   }%
583   \fi
584   \fi
585   \let\AskInc@@QuestionHeader\AskInc@QuestionHeader
586   \AskInc@AskQuestion
587   \ifx\AskInc@AskResult\AskInc@CharYes
588     \ifx\AskInc@Answer\empty
589       \gdef\AskInc@Answer{\#1}%
590     \else
591       \g@addto@macro\AskInc@Answer{,\#1}%
592     \fi
593   \AskInc@SaveAnswer
594   \let\@partlist\AskInc@Answer
595   \fi
596 }

\AskInc@AskQuestion
597 \def\AskInc@AskQuestion{%
598   \let\AskInc@Temp\answer
599   \AskInc@@QuestionHeader
600   \let\AskInc@@QuestionHeader\relax
601   \typein[\answer]{%
602     Include '\AskInc@File'? [y]es, [n]o, [A]ll, [N]one, [D]efault %
603     (\AskInc@Default):%
604   }%

```

```

605  \typeout{ }%
606  \let\AskInc@AskResult\answer
607  \let\answer\AskInc@Temp
608  \ifx\AskInc@AskResult\empty
609    \let\AskInc@AskResult\AskInc@Default
610    \let\AskInc@Next\relax
611  \else
612    \ifx\AskInc@AskResult\AskInc@CharYes
613      \let\AskInc@Next\relax
614    \else
615      \ifx\AskInc@AskResult\AskInc@CharNo
616        \let\AskInc@Next\relax
617      \else
618        \ifx\AskInc@AskResult\AskInc@CharAll
619          \gdef\AskInc@AskQuestion{ }
620          \let\AskInc@AskResult\AskInc@CharYes
621        }%
622      \else
623        \ifx\AskInc@AskResult\AskInc@CharNone
624          \gdef\AskInc@AskQuestion{ }
625          \let\AskInc@AskResult\AskInc@CharNo
626        }%
627      \else
628        \ifx\AskInc@AskResult\AskInc@CharDefault
629          \gdef\AskInc@AskQuestion{ }
630          \let\AskInc@AskResult\AskInc@Default
631        }%
632      \else
633        \PackageWarning{askinlude}{%
634          Invalid answer [\AskInc@AskResult].\MessageBreak
635          Answer must be one letter out of [ynAND]%
636        }%
637      \fi
638    \fi
639  \fi
640  \let\AskInc@Next\AskInc@AskQuestion
641 \fi
642 \fi
643 \fi
644 \AskInc@Next
645 }

646 \AtBeginDocument{\AskInc@MainQuestion}
647 \AskInc@AtEnd%
648 
```

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

CTAN:macros/latex/contrib/askinlude/askinlude.dtx The source file.

CTAN:macros/latex/contrib/askinlude/askinlude.pdf Documentation.

¹CTAN:pkg/askinlude

Bundle. All the packages of the bundle ‘askininclude’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:install/macros/latex/contrib/askininclude.tds.zip](#)

TDS refers to the standard “A Directory Structure for TeX Files” ([CTAN:pkg/tds](#)). Directories with `texmf` in their name are usually organized this way.

3.2 Bundle installation

Unpacking. Unpack the `askininclude.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip askininclude.tds.zip -d ~/texmf
```

3.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain TeX:

```
tex askininclude.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
askininclude.sty → tex/latex/askininclude/askininclude.sty  
askininclude.pdf → doc/latex/askininclude/askininclude.pdf  
askininclude.dtx → source/latex/askininclude/askininclude.dtx
```

If you have a `docstrip.cfg` that configures and enables `docstrip`’s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

3.4 Refresh file name databases

If your TeX distribution (TeX Live, mikTeX, ...) relies on file name databases, you must refresh these. For example, TeX Live users run `texhash` or `mktexlsr`.

3.5 Some details for the interested

Unpacking with L^AT_EX. The `.dtx` chooses its action depending on the format:

plain TeX: Run `docstrip` and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for `docstrip` (really, `docstrip` does not need L^AT_EX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{askininclude.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfL^AT_EX:

```
pdflatex askinlude.dtx
bibtex askinlude.aux
makeindex -s gind.ist askinlude.idx
pdflatex askinlude.dtx
makeindex -s gind.ist askinlude.idx
pdflatex askinlude.dtx
```

4 Acknowledgement

Tom Scavo suggested saving the include file list from the previous run and a simplified user interface.

Joseph Wright suggested questions for each `\include`.

5 References

- [1] Robin Fairbairns. *[tex-live] licence on includeonly.sty*. 2007-10-13. URL: <https://www.tug.org/pipermail/tex-live/2007-October/014928.html> (visited on 2011-12-02).
- [2] Daniel H. Luecking and Donald Arseneau. *The excludeonly package*. 2003-03-14. URL: <https://ctan.org/pkg/excludeonly> (visited on 2011-12-02).
- [3] Herbert Voß. *askinlude.sty*. Thread in newsgroup `comp.text.tex`; Message-ID: `fev72m$b35$02$2@news.t-online.com`. 2007-10-15. URL: https://groups.google.com/group/comp.text.tex/browse_thread/thread/728a3e73c04c6789/349e31865c40d00c (visited on 2011-12-02).

6 History

[1991/11/05 v0.95]

- First beta-test version (asked two questions).

[1991/11/05 v0.99]

- Second beta-test version (asks one question; uses ‘*’ and ‘-’).

[1991/11/07 v1.0]

- Changed prompts (sent to `comp.text.tex`): “Interactive inclusion of files in L^AT_EX”²

[1991/12/13 v1.1]

- Unknown.

²Url: <https://groups.google.com/group/comp.text.tex/msg/34b770a7c277e79c>

[1994/11/02 v1.2e]

- Two small changes to run under LaTeX2e [RF].
- Additional documentation [TRS].

[2007/10/15 v1.0 (askinc)]

- Reimplementation, published as `askinc` in newsgroup `comp.text.tex`:
“Re: `askinclude.sty`”³
- Listing of included files, collected from the previous run before the question.
- No error message by `.aux` file if package is removed.

[2007/10/23 v2.0]

- The next version of `askinc` becomes version 2.0 of `askinclude`.
- ‘?’ added for questioning each file.
- `\includeonly` has precedence until package loading. Then it is disabled by a warning.
- Documentation rewritten/added.
- Tests added.

[2008/08/11 v2.1]

- Code is not changed.
- URLs updated.

[2011/12/02 v2.2]

- Bibliography rewritten for biblatex.

[2016/05/14 v2.3]

- Luatex compatibility

[2018/11/01 v2.4]

- Remove luatex85 package dependency

[2019/12/09 v2.5]

- Documentation updates.

[2019-12-10 v2.6]

- Updated

³Url: <https://groups.google.com/group/comp.text.tex/msg/2d638b84a858fdc8>

7 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; plain numbers refer to the code lines where the entry is used.

Symbols	
\!	361
\\$	226, 363
\(219, 364
\)	220, 365
*	221, 357
\+	222, 366
\,	360
\-	358
\.	229, 372
\:	371
\?	223, 359
\empty	80, 142, 145, 158, 201, 232, 244, 253, 257, 268, 269, 279, 295, 311, 323, 380, 393, 396, 407, 408, 414, 417, 421, 441, 479, 506, 514, 553, 563, 588, 608
\firstofone	61, 462, 473
\firstoftwo	280, 564
\for	272, 284, 285, 568, 569, 578
\gobble	107, 118, 119, 462, 475, 536
\ifundefined	128, 133, 156, 237, 243, 247, 256, 549, 552
\includeonly	140
\mainaux	67
\makeother	357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372
\nil	479, 481
\partlist	441, 442, 444, 594
\secondoftwo	282, 566
\spaces	293, 318, 338, 344, 346, 347, 348, 374
\undefined	54, 76, 453, 464, 523
\[217, 367
\\	216, 369
\{	227
\}	228
\]	218, 368
\^	225, 362
\	224, 370
A	
\answer	373, 376, 598, 601, 606, 607
\AskInc@@Answer	376, 379, 382, 385, 396, 397, 406, 412, 413, 416
\AskInc@@CheckMatch	479, 481
\AskInc@@QuestionHeader	585, 599, 600
\AskInc@AddFile	99, 106, 112
\AskInc@AddList	75, 100, 529
\AskInc@AddMatch	528, 535, 541
\AskInc@Answer	130, 141, 142, 143, 238, 240, 242, 273, 277, 284, 318, 397, 407, 412, 423, 436, 546, 588, 589, 591, 594
\AskInc@AskFile	118, 122, 409
\AskInc@AskQuestion	586, 597
\AskInc@AskResult	587, 606, 608, 609, 612, 615, 618, 620, 623, 625, 628, 630, 634
\AskInc@AtBeginDocument	60, 65
\AskInc@AtEnd	29, 30, 51, 165, 647
\AskInc@CharAll	211, 618
\AskInc@CharAsterisk	204, 238, 273, 413, 550, 557
\AskInc@CharDefault	213, 628
\AskInc@CharExclam	208, 483
\AskInc@CharHash	72, 76, 78, 80, 87, 89
\AskInc@CharMinus	205, 276, 277, 416, 561
\AskInc@CharNo	210, 560, 576, 580, 615, 625
\AskInc@CharNone	212, 623
\AskInc@CharPlus	206, 274, 288, 294
\AskInc@CharQuestion	207, 406
\AskInc@CharYes	209, 558, 572, 577, 587, 612, 620
\AskInc@CheckMatch	471, 478
\AskInc@Default	558, 560, 572, 576, 577, 580, 603, 609, 630
\AskInc@DoAskFile	409, 555
\AskInc@DoInclude	432
\AskInc@DoMatch	439, 457, 468
\AskInc@File	272, 287, 293, 296, 300, 556, 571, 579, 602
\AskInc@Files	100, 115, 269, 272, 578
\AskInc@FilterPattern	508, 510
\AskInc@Foundfalse	469, 486
\AskInc@Foundtrue	493
\AskInc@Include	432
\AskInc@InvalidPattern	488, 495, 499
\AskInc@MainAnswer	399, 405
\AskInc@MainQuestion	350, 646
\AskInc@MakeMatchInit	449, 455
\AskInc@Mark	274, 276, 288, 293, 294
\AskInc@MatchFile	119, 123, 438
\AskInc@MatchFiles	161, 257, 285, 529, 569
\AskInc@MatchInit	426, 434
\AskInc@MatchList	437, 461, 471, 505, 508, 514, 515, 517

\AskInc@MatchMethod	135, 145, 158, 244, 246, 248, 279, 311, 314, 408, 414, 417, 420, 448	D
\AskInc@MatchOption	186, 190, 201, 248, 250, 253, 338, 380, 383, 387, 393, 420, 421	\DeclareOption 152, 153, 154, 185, 189, 193, 200
\AskInc@Next	378, 399, 403, 610, 613, 616, 640, 644	\do 272, 286, 570, 578
\AskInc@OrgInclude	116, 124	E
\AskInc@OrgIncludeOnly	167, 410, 418, 422, 427	\endcsname 10, 181
\AskInc@Pattern	511, 512, 515, 518	\endinput 51
\AskInc@PdfMatchAvailabletrue	183	\endlinechar 5, 11, 23
\AskInc@PdfMatchInit	451, 466	\escapechar 216
\AskInc@PreviousAnswer	242, 549, 557, 561, 568	G
\AskInc@PreviousMatches	268, 295, 296, 299	\g@addto@macro 89, 298, 444, 517, 591
\AskInc@PreviousMatchMethod	246, 552, 563	\gdef 87, 130, 135, 141, 143, 161, 442, 546, 589, 619, 624, 629
\AskInc@QuestionHeader	260, 351, 585	I
\AskInc@SanitizeAnswer	214, 240, 376	\if@filesw 66
\AskInc@SaveAnswer	544, 593	\if@partsw 147
\AskInc@StringMakeMatch	177, 186, 326, 382, 383, 448	\ifAskInc@Found 467, 472
\AskInc@StringNoRegExp	176, 379	\ifAskInc@PdfMatchAvailable 179, 188, 251, 329, 386
\AskInc@StringPdfMatch	178, 190, 250, 330, 385, 387	\ifcase 76, 484, 491
\AskInc@Temp	284, 285, 287, 458, 459, 461, 462, 482, 483, 500, 502, 504, 512, 568, 569, 571, 578, 579, 598, 607	\ifx 54, 76, 78, 80, 142, 158, 181, 250, 269, 273, 277, 279, 287, 294, 295, 311, 338, 379, 382, 385, 396, 406, 413, 416, 421, 441, 448, 483, 512, 514, 557, 561, 563, 571, 576, 579, 587, 588, 608, 612, 615, 618, 623, 628
\AskInc@TypeoutFiles	267, 352	\immediate 67
\AskInc@TypeoutMainQuestion	342, 355	\include 117, 120, 271
\AskInc@TypeoutMatchMethod	321, 354	\includeonly 149, 168, 169
\AskInc@TypeoutMethod	323, 326, 330, 336	M
\AskInc@TypeoutPreviousAnswer	308, 353	\MakeMatcher 459
\AskInc@WriteAddFile	109, 121	\MessageBreak 149, 172, 195, 390, 634
\AskInc@WriteAddMatch	440, 538	N
\AskInc@WriteAux	64, 74, 98, 111, 129, 134, 160, 527, 540, 545	\NeedsTeXFormat 2
\AskInc@WriteDefAddFile	96, 110	\newcommand 116, 167
\AskInc@WriteDefAddList	73, 97, 526	\newif 179, 467
\AskInc@WriteDefAddMatch	525, 539	\numexpr 55
\AtBeginDocument	... 60, 63, 106, 127, 236, 535, 646	O
\AtEndOfPackage	126	\outputmode 58
C	P	
\catcode	3, 4, 6, 7, 8, 12, 13, 14, 15, 16, 17, 18, 21, 22, 24, 25, 26, 27, 31, 33	\PackageWarning 170, 389
\comma@list	437	\PackageWarningNoLine 148, 194, 501, 633
\comma@normalize	436	\pdfextension 54, 56, 57
\comma@parse@normalized	470, 507	\pdffeedback 55
\csname	10, 181	\pdflastobj 55
		\pdfmatch 196, 331, 391, 484, 491, 502
		\pdfnames 56
		\pdfobj 57
		\pdfoutput 58
		\ProcessOptions 155, 203
		\protected 55, 56, 57
		\providecommand 63, 75, 99, 115, 140, 528

<pre>\ProvidesPackage 52 R \renewcommand 120, 169 \RequirePackage 435, 456</pre>	<pre>\space 149, S</pre>	<pre>\typein 373, 601 \typeout 261, 262, W</pre>
		<pre>263, 264, 265, 271, 293, 305, X</pre>
		<pre>309, 318, 319, 322, 334, 337, \write 67 Z</pre>
		<pre>343, 344, 346, 347, 348, 375, 605 \the 11, 12, 13, 14, 15, 16, 17, 18, 31 \TMP@EnsureCode 28, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50 \zap@space 231</pre>