

The askinclude package

Pablo A. Straub

Heiko Oberdiek (maintainer)

<heiko.oberdiek at googlegmail.com>

2011/12/02 v2.2

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> | 9 |
| 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> | 10 |
| 2.10 | Options | 11 |
| 2.11 | Strings | 11 |
| 2.12 | Set defaults | 12 |
| 2.13 | Questions | 13 |
| 2.13.1 | Generic question header | 13 |
| 2.13.2 | Display files | 13 |
| 2.13.3 | Previous answer | 14 |
| 2.13.4 | Current regular expression support | 14 |
| 2.13.5 | Main question text | 14 |
| 2.13.6 | Main question | 15 |
| 2.14 | Evaluate answer of main question | 15 |
| 2.15 | Ask for each file | 19 |

| | | |
|----------|--|-----------|
| 3 | Test | 20 |
| 3.1 | Included files | 20 |
| 3.2 | Special test setup | 21 |
| 3.3 | Common start | 21 |
| 3.4 | First run tests | 24 |
| 3.5 | Second run tests | 28 |
| 3.6 | Compatibility for changing from old to new | 32 |
| 3.7 | Regular expressions | 34 |
| 3.8 | Common end | 35 |
| 4 | Installation | 35 |
| 4.1 | Download | 35 |
| 4.2 | Bundle installation | 35 |
| 4.3 | Package installation | 35 |
| 4.4 | Refresh file name databases | 36 |
| 4.5 | Some details for the interested | 36 |
| 5 | Catalogue | 37 |
| 6 | Acknowledgement | 37 |
| 7 | References | 37 |
| 8 | History | 38 |
| | [1991/11/05 v0.95] | 38 |
| | [1991/11/05 v0.99] | 38 |
| | [1991/11/07 v1.0] | 38 |
| | [1991/12/13 v1.1] | 38 |
| | [1994/11/02 v1.2e] | 38 |
| | [2007/10/15 v1.0 (askinc)] | 38 |
| | [2007/10/23 v2.0] | 38 |
| | [2008/08/11 v2.1] | 39 |
| | [2011/12/02 v2.2] | 39 |
| 9 | Index | 39 |

1 Documentation

1.1 User interface

The `\include` feature of \LaTeX 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{askinclude}

% 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 `askinclude` 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 askinclude 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 1.2](#).

`[]`: Just pressing return (empty string) uses the default answer, that was stored in the auxiliary file (`.aux`) of the previous `LATEX`run with package `askinclude`. 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 pdf \TeX 1.30 or later is used. It relies on an undocumented feature `\pdfmatch` that may change in future versions of pdf \TeX .

Package `askinclude` 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 askinclude 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 machtes 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 `askinclude` 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 reimplementaion 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 `askinclude` 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 reimplementing 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_ε. 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{askinclude}%
53 [2011/12/02 v2.2 Interactive asking of included files]%

```

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.

```

54 \AtBeginDocument{%
55   \let\AskInc@AtBeginDocument\@firstofone
56 }
57 \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.

```

58 \def\AskInc@WriteAux#1#2{%
59   \AskInc@AtBeginDocument{%
60     \if@files
61       \immediate\write\@mainaux{#1}%
62       #2%
63     \fi
64   }%
65 }

```

2.4 Remember \include files

`\AskInc@CharHash`

```

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

```

`\AskInc@WriteDefAddList`

```

67 \def\AskInc@WriteDefAddList{%
68   \AskInc@WriteAux{%

```

`\AskInc@AddList`

```

69     \string\providecommand*\string\AskInc@AddList[2]{%
70       \string\ifcase\string\ifx\AskInc@CharHash1\string\@undefined
71         \string\else
72           \string\ifx\AskInc@CharHash1\string\relax
73           \string\else
74             \string\ifx\AskInc@CharHash1\string\@empty
75             \string\else
76               1%
77             \string\fi
78           \string\fi
79         \string\fi
80         0 %
81       \string\gdef\AskInc@CharHash1{\AskInc@CharHash2}%
82     \string\else
83       \string\g@addto@macro\AskInc@CharHash1{\AskInc@CharHash2}%

```



```

84     \string\fi
85   }%

86 }{%
87   \global\let\AskInc@WriteDefAddList\relax
88 }%
89 }

\AskInc@WriteDefAddFile
90 \def\AskInc@WriteDefAddFile{%
91   \AskInc@WriteDefAddList
92   \AskInc@WriteAux{%

\AskInc@AddFile
93     \string\providecommand\string\AskInc@AddFile{%
94       \string\AskInc@AddList\string\AskInc@Files
95     }%

96 }{%
97   \global\let\AskInc@WriteDefAddFile\relax
98 }%
99 }

\AskInc@AddFile
100 \AtBeginDocument{%
101   \global\let\AskInc@AddFile\@gobble
102 }

\AskInc@WriteAddFile
103 \def\AskInc@WriteAddFile#1{%
104   \AskInc@WriteDefAddFile
105   \AskInc@WriteAux{%
106     \string\AskInc@AddFile{#1}%
107   }{%
108 }

\AskInc@Files
109 \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
110 \newcommand{\AskInc@OrgInclude}{}
111 \let\AskInc@OrgInclude\include

\AskInc@AskFile
112 \let\AskInc@AskFile\@gobble

\AskInc@MatchFile
113 \let\AskInc@MatchFile\@gobble

\include
114 \renewcommand*\include[1]{%
115   \AskInc@WriteAddFile{#1}%
116   \AskInc@AskFile{#1}%
117   \AskInc@MatchFile{#1}%
118   \AskInc@OrgInclude{#1}%
119 }

```

2.6 Store answer and match method in .aux file

```
120 \AtEndOfPackage{%
121   \AtBeginDocument{%
122     \ifundefined{AskInc@Answer}{-}{%
123       \AskInc@WriteAux{%
124         \string\gdef\string\AskInc@Answer{\AskInc@Answer}%
125       }{}%
126     }%
127     \ifundefined{AskInc@MatchMethod}{-}{%
128       \AskInc@WriteAux{%
129         \string\gdef\string\AskInc@MatchMethod{\AskInc@MatchMethod}%
130       }{}%
131     }%
132   }%
133 }
```

2.7 Support for \@includeonly

`\@includeonly` Macro `\@includeonly` is defined for compatibility reasons only, if the .aux file was written by versions before 2 of package askinclude.

```
134 \providecommand*\@includeonly}[1]{%
135   \gdef\AskInc@Answer{#1}%
136   \ifx\AskInc@Answer\@empty
137     \gdef\AskInc@Answer{-}%
138   \fi
139   \global\let\AskInc@MatchMethod\@empty
140 }
```

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.

```
141 \if@partsw
142   \PackageWarningNoLine{askinclude}{%
143     Previous \string\includeonly\space detected,\MessageBreak
144     therefore asking for files is suppressed%
145   }%
146 \fi
```

Dummy options.

```
146 \DeclareOption{makematch}{-}%
147 \DeclareOption{pdfmatch}{-}%
148 \DeclareOption{noregexp}{-}%
149 \ProcessOptions*
```

Preserve `\AskInc@MatchFiles`.

```
150 \ifundefined{AskInc@MatchFiles}{%
151   -}%
152   \ifx\AskInc@MatchMethod\@empty
153   \else
154     \AskInc@WriteAux{%
155       \string\gdef\string\AskInc@MatchFiles{\AskInc@MatchFiles}%
156     }{}%
157   \fi
158 }%
159 \expandafter\AskInc@AtEnd
160 \fi%
```

2.9 \includeonly

`\AskInc@OrgIncludeOnly`

```
161 \newcommand{\AskInc@OrgIncludeOnly}{-}%
162 \let\AskInc@OrgIncludeOnly\includeonly
```

\includeonly

```
163 \renewcommand*{\includeonly}[1]{%
164   \PackageWarning{askinclude}{%
165     \string\includeonly\space is managed by package `askinclude',%
166     \MessageBreak
167     therefore it is ignored%
168   }%
169 }
```

2.10 Options

\AskInc@StringNoRegExp

```
170 \def\AskInc@StringNoRegExp{noregexp}
```

\AskInc@StringMakeMatch

```
171 \def\AskInc@StringMakeMatch{makematch}
```

\AskInc@StringPdfMatch

```
172 \def\AskInc@StringPdfMatch{pdfmatch}
```

\ifAskInc@PdfMatchAvailable

```
173 \newif\ifAskInc@PdfMatchAvailable
174 \begingroup\expandafter\expandafter\expandafter\endgroup
175 \expandafter\ifx\csname pdfmatch\endcsname\relax
176 \else
177   \AskInc@PdfMatchAvailabletrue
178 \fi

179 \DeclareOption{makematch}{%
180   \let\AskInc@MatchOption\AskInc@StringMakeMatch
181 }
182 \ifAskInc@PdfMatchAvailable
183   \DeclareOption{pdfmatch}{%
184     \let\AskInc@MatchOption\AskInc@StringPdfMatch
185   }%
186 \else
187   \DeclareOption{pdfmatch}{%
188     \PackageWarningNoLine{askinclude}{%
189       Option `pdfmatch' is not available because\MessageBreak
190       your pdfTeX does not provide \string\pdfmatch
191     }%
192   }%
193 \fi
194 \DeclareOption{noregexp}{%
195   \let\AskInc@MatchOption\@empty
196 }
197 \ProcessOptions*
```

2.11 Strings

\AskInc@CharAsterisk

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

\AskInc@CharMinus

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

\AskInc@CharPlus

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

\AskInc@CharQuestion

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

```

\AskInc@CharExclam
202 \def\AskInc@CharExclam{!}

\AskInc@CharYes
203 \def\AskInc@CharYes{y}

\AskInc@CharNo
204 \def\AskInc@CharNo{n}

\AskInc@CharAll
205 \def\AskInc@CharAll{A}

\AskInc@CharNone
206 \def\AskInc@CharNone{N}

\AskInc@CharDefault
207 \def\AskInc@CharDefault{D}

```

2.12 Set defaults

```

\AskInc@SanitizeAnswer
208 \def\AskInc@SanitizeAnswer#1#2{%
209   \begingroup
210   \escapechar=`\%
211   \edef\[\{\string\[%
212   \edef\]\}\string\}%
213   \edef\(\{\string\(%
214   \edef\)\}\string\)%
215   \edef\*\{\string\*}%
216   \edef\+\{\string\+}%
217   \edef\?\{\string\?}%
218   \edef\|\{\string\|}%
219   \edef\^\{\string\^}%
220   \edef\$\{\string\$}%
221   \edef\{\{\string\{%
222   \edef\}\}\string\}%
223   \edef\.\{\string\.}%
224   \xdef#1{%
225     \expandafter\expandafter\expandafter\zap@space
226     \expandafter#2\space\@empty
227   }%
228 \endgroup
229 }

230 \AtBeginDocument{%
231   \ifundefined{AskInc@Answer}{%
232     \global\let\AskInc@Answer\AskInc@CharAsterisk
233   }{%
234     \AskInc@SanitizeAnswer\AskInc@Answer\AskInc@Answer
235   }%
236   \global\let\AskInc@PreviousAnswer\AskInc@Answer
237   \ifundefined{AskInc@MatchMethod}{%
238     \global\let\AskInc@MatchMethod\@empty
239   }{%
240     \global\let\AskInc@PreviousMatchMethod\AskInc@MatchMethod
241     \ifundefined{AskInc@MatchOption}{%
242       \let\AskInc@MatchOption\AskInc@MatchMethod
243     }{%
244       \ifx\AskInc@MatchOption\AskInc@StringPdfMatch
245         \ifAskInc@PdfMatchAvailable
246         \else

```

```

247     \let\AskInc@MatchOption\@empty
248   \fi
249 \fi
250 \@ifundefined{AskInc@MatchFiles}{%
251   \let\AskInc@MatchFiles\@empty
252 }{}%
253 }

```

2.13 Questions

2.13.1 Generic question header

\AskInc@QuestionHeader

```

254 \def\AskInc@QuestionHeader{%
255   \typeout{}%
256   \typeout{*****}%
257   \typeout{*** Package askinclude Question ***}%
258   \typeout{*****}%
259   \typeout{}%
260 }

```

2.13.2 Display files

\AskInc@TypeoutFiles

```

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

```

```

299 \typeout{%
300 \fi
301 }

```

2.13.3 Previous answer

\AskInc@TypeoutPreviousAnswer

```

302 \def\AskInc@TypeoutPreviousAnswer{%
303 \typeout{%
304 Previous answer (%
305 \ifx\AskInc@MatchMethod\@empty
306 noregexp%
307 \else
308 \AskInc@MatchMethod
309 \fi
310 ):%
311 }%
312 \typeout{ \@spaces[\AskInc@Answer]]}%
313 \typeout{}}%
314 }

```

2.13.4 Current regular expression support

\AskInc@TypeoutMatchMethod

```

315 \def\AskInc@TypeoutMatchMethod{%
316 \typeout{Regular expressions:}%
317 \AskInc@TypeoutMethod\@empty{noregexp}{%
318 \space disabled%
319 }%
320 \AskInc@TypeoutMethod\AskInc@StringMakeMatch{makematch}{%
321 enabled, using package `makematch'%
322 }%
323 \ifAskInc@PdfMatchAvailable
324 \AskInc@TypeoutMethod\AskInc@StringPdfMatch{pdfmatch}{%
325 \space enabled, using pdfTeX's \string\pdfmatch
326 }%
327 \fi
328 \typeout{}}%
329 }

```

\AskInc@TypeoutMethod

```

330 \def\AskInc@TypeoutMethod#1#2#3{%
331 \typeout{%
332 \ifx#1\AskInc@MatchOption \space--> \else\@spaces\space\fi
333 [#2] #3% hash-ok
334 }%
335 }

```

2.13.5 Main question text

\AskInc@TypeoutMainQuestion

```

336 \def\AskInc@TypeoutMainQuestion{%
337 \typeout{Which files do you want to include?}%
338 \typeout{ \@spaces[foo,bar] \space\space
339 comma separated file or pattern list}%
340 \typeout{ \@spaces[*] \@spaces\@spaces all files}%
341 \typeout{ \@spaces[-] \@spaces\@spaces no files}%
342 \typeout{ \@spaces[?] \@spaces\@spaces ask for each file}%
343 }

```

2.13.6 Main question

\AskInc@MainQuestion

```

344 \def\AskInc@MainQuestion{%
345   \AskInc@QuestionHeader
346   \AskInc@TypeoutFiles
347   \AskInc@TypeoutPreviousAnswer
348   \AskInc@TypeoutMatchMethod
349   \AskInc@TypeoutMainQuestion
350   \begingroup
351     \@makeother\*%
352     \@makeother\-%
353     \@makeother\?%
354     \@makeother\,%
355     \@makeother\!%
356     \@makeother\~%
357     \@makeother\$%
358     \@makeother\(%
359     \@makeother\)%
360     \@makeother\+%
361     \@makeother\[%
362     \@makeother\]%
363     \@makeother\\%
364     \@makeother\|%
365     \@makeother\:%
366     \@makeother\.%
367     \typein[\answer]%
368     { \@spaces[] \space\@spaces\@spaces use the previous answer}%
369     \typeout{}%
370     \AskInc@SanitizeAnswer\AskInc@@Answer\answer
371   \endgroup
372   \let\AskInc@Next\AskInc@MainQuestion
373   \ifx\AskInc@@Answer\AskInc@StringNoRegExp
374     \let\AskInc@MatchOption\@empty
375   \else
376     \ifx\AskInc@@Answer\AskInc@StringMakeMatch
377       \let\AskInc@MatchOption\AskInc@StringMakeMatch
378     \else
379       \ifx\AskInc@@Answer\AskInc@StringPdfMatch
380         \ifAskInc@PdfMatchAvailable
381           \let\AskInc@MatchOption\AskInc@StringPdfMatch
382         \else
383           \PackageWarning{askinclude}{%
384             Method `pdfmatch' is not available because\MessageBreak
385             your pdfTeX does not provide \string\pdfmatch
386           }%
387           \let\AskInc@MatchOption\@empty
388         \fi
389       \else
390         \ifx\AskInc@@Answer\@empty
391           \global\let\AskInc@@Answer\AskInc@Answer
392         \fi
393         \let\AskInc@Next\AskInc@MainAnswer
394       \fi
395     \fi
396   \fi
397   \AskInc@Next
398 }

```

2.14 Evaluate answer of main question

\AskInc@MainAnswer

```

399 \def\AskInc@MainAnswer{%
400   \ifx\AskInc@@Answer\AskInc@CharQuestion
401     \global\let\AskInc@Answer@empty
402     \global\let\AskInc@MatchMethod@empty
403     \global\let\AskInc@AskFile\AskInc@DoAskFile
404     \AskInc@OrgIncludeOnly}%
405   \else
406     \global\let\AskInc@Answer\AskInc@@Answer
407     \ifx\AskInc@@Answer\AskInc@CharAsterisk
408       \global\let\AskInc@MatchMethod@empty
409       \else
410         \ifx\AskInc@@Answer\AskInc@CharMinus
411           \global\let\AskInc@MatchMethod@empty
412           \AskInc@OrgIncludeOnly}%
413         \else
414           \global\let\AskInc@MatchMethod\AskInc@MatchOption
415           \ifx\AskInc@MatchOption@empty
416             \expandafter\AskInc@OrgIncludeOnly\expandafter{%
417               \AskInc@Answer
418             }%
419           \else
420             \AskInc@MatchInit
421             \AskInc@OrgIncludeOnly}%
422           \fi
423         \fi
424       \fi
425     \fi
426   \let\AskInc@Include\AskInc@DoInclude
427 }

```

\AskInc@MatchInit

```

428 \def\AskInc@MatchInit{%
429   \RequirePackage{kvsetkeys}[2007/09/29]%
430   \expandafter\comma@normalize\expandafter{\AskInc@Answer}%
431   \global\let\AskInc@MatchList\comma@list

```

\AskInc@MatchFile

```

432 \def\AskInc@MatchFile##1{%
433   \AskInc@DoMatch{##1}{%
434     \AskInc@WriteAddMatch{##1}%
435     \ifx\@partlist@empty
436       \gdef\@partlist{##1}%
437     \else
438       \g@addto@macro\@partlist{,##1}%
439     \fi
440   }%
441 }%

442 \ifx\AskInc@MatchMethod\AskInc@StringMakeMatch
443   \AskInc@MakeMatchInit
444 \else
445   \AskInc@PdfMatchInit
446 \fi
447 \let\AskInc@MatchInit\undefined
448 }

```

\AskInc@MakeMatchInit

```

449 \def\AskInc@MakeMatchInit{%
450   \RequirePackage{makematch}%

```

\AskInc@DoMatch

```

451 \def\AskInc@DoMatch##1{%

```



```

452 \def\AskInc@Temp{%
453 \MakeMatcher[,]\AskInc@Temp
454 }%
455 \expandafter\AskInc@Temp\expandafter{\AskInc@MatchList}%
456 \AskInc@Temp{##1}\@firstofone\@gobble
457 }%

458 \let\AskInc@MakeMatchInit\@undefined
459 }

```

\AskInc@PdfMatchInit

```

460 \def\AskInc@PdfMatchInit{%

\ifAskInc@Found

461 \newif\ifAskInc@Found

\AskInc@DoMatch

462 \def\AskInc@DoMatch##1{%
463 \AskInc@Foundfalse
464 \expandafter\comma@parse@normalized\expandafter
465 {\AskInc@MatchList}\AskInc@CheckMatch{##1}}%
466 \ifAskInc@Found
467 \expandafter\@firstofone
468 \else
469 \expandafter\@gobble
470 \fi
471 }%

```

\AskInc@CheckMatch

```

472 \def\AskInc@CheckMatch##1##2{%
473 \AskInc@@CheckMatch##2\@empty\@nil{##1}%
474 }%

```

\AskInc@@CheckMatch

```

475 \def\AskInc@@CheckMatch##1##2\@nil##3{%
476 \def\AskInc@Temp{##1}%
477 \ifx\AskInc@Temp\AskInc@CharExclam
478 \ifcase\pdfmatch{##2}{##3}=1 %
479 \or
480 \AskInc@Foundfalse
481 \else
482 \AskInc@InvalidPattern{##2}{##1##2}%
483 \fi
484 \else
485 \ifcase\pdfmatch{##1##2}{##3} %
486 \or
487 \AskInc@Foundtrue
488 \else
489 \AskInc@InvalidPattern{##1##2}{##1##2}%
490 \fi
491 \fi
492 }%

```

\AskInc@InvalidPattern Prints a warning with the regular expression and removes it from the list to prevent further warnings for the same expression.

```

493 \def\AskInc@InvalidPattern##1##2{%
494 \edef\AskInc@Temp{##1}%
495 \PackageWarningNoLine{askinclude}{%
496 Invalid pattern '\AskInc@Temp' for \string\pdfmatch
497 }%
498 \edef\AskInc@Temp{##2}%
499 \global\expandafter\let\expandafter\AskInc@MatchList

```

```

500     \expandafter\@empty
501     \expandafter\comma@parse@normalized\expandafter
502     {\AskInc@MatchList}\AskInc@FilterPattern
503 }%

\AskInc@FilterPattern

504 \def\AskInc@FilterPattern##1{%
505     \edef\AskInc@Pattern{##1}%
506     \ifx\AskInc@Temp\AskInc@Pattern
507     \else
508         \ifx\AskInc@MatchList\@empty
509             \global\let\AskInc@MatchList\AskInc@Pattern
510         \else
511             \expandafter\g@addto@macro\expandafter\AskInc@MatchList{%
512                 \expandafter,\AskInc@Pattern
513             }%
514         \fi
515     \fi
516 }%

517 \let\AskInc@PdfMatchInit\@undefined
518 }

\AskInc@WriteDefAddMatch

519 \def\AskInc@WriteDefAddMatch{%
520     \AskInc@WriteDefAddList
521     \AskInc@WriteAux{%

\AskInc@AddMatch

522     \string\providecommand\string\AskInc@AddMatch{%
523         \string\AskInc@AddList\string\AskInc@MatchFiles
524     }%

525 }{%
526     \global\let\AskInc@WriteDefAddMatch\relax
527 }%
528 }

\AskInc@AddMatch

529 \AtBeginDocument{%
530     \global\let\AskInc@AddMatch\@gobble
531 }

\AskInc@WriteAddMatch

532 \def\AskInc@WriteAddMatch#1{%
533     \AskInc@WriteDefAddMatch
534     \AskInc@WriteAux{%
535         \string\AskInc@AddMatch{#1}%
536     }{%
537 }

\AskInc@SaveAnswer

538 \def\AskInc@SaveAnswer{%
539     \AskInc@WriteAux{%
540         \string\gdef\string\AskInc@Answer{\AskInc@Answer}%
541     }{%
542 }

```

2.15 Ask for each file

\AskInc@PreviousAnswer

```
543 \@ifundefined{AskInc@PreviousAnswer}{%
544   \let\AskInc@PreviousAnswer\AskInc@CharAsterisk
545 }{}
```

\AskInc@PreviousMatchMethod

```
546 \@ifundefined{AskInc@PreviousMatchMethod}{%
547   \let\AskInc@PreviousMatchMethod\@empty
548 }{}
```

\AskInc@DoAskFile

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

\AskInc@AskQuestion

```
591 \def\AskInc@AskQuestion{%
592   \let\AskInc@Temp\answer
593   \AskInc@@QuestionHeader
594   \let\AskInc@@QuestionHeader\relax
595   \typein[\answer]{%
```

```

596 Include `AskInc@File'? [y]es, [n]o, [A]ll, [N]one, [D]efault %
597 (\AskInc@Default):%
598 }%
599 \typeout{%
600 \let\AskInc@AskResult\answer
601 \let\answer\AskInc@Temp
602 \ifx\AskInc@AskResult\@empty
603   \let\AskInc@AskResult\AskInc@Default
604   \let\AskInc@Next\relax
605 \else
606   \ifx\AskInc@AskResult\AskInc@CharYes
607     \let\AskInc@Next\relax
608   \else
609     \ifx\AskInc@AskResult\AskInc@CharNo
610       \let\AskInc@Next\relax
611     \else
612       \ifx\AskInc@AskResult\AskInc@CharAll
613         \gdef\AskInc@AskQuestion{%
614           \let\AskInc@AskResult\AskInc@CharYes
615         }%
616       \else
617         \ifx\AskInc@AskResult\AskInc@CharNone
618           \gdef\AskInc@AskQuestion{%
619             \let\AskInc@AskResult\AskInc@CharNo
620           }%
621         \else
622           \ifx\AskInc@AskResult\AskInc@CharDefault
623             \gdef\AskInc@AskQuestion{%
624               \let\AskInc@AskResult\AskInc@Default
625             }%
626           \else
627             \PackageWarningNoLine{askinclude}{%
628               Invalid answer [\AskInc@AskResult].\MessageBreak
629               Answer must be one letter out of [ynAND]%
630             }%
631           \fi
632         \fi
633       \fi
634       \let\AskInc@Next\AskInc@AskQuestion
635     \fi
636   \fi
637 \fi
638 \AskInc@Next
639 }

640 \AtBeginDocument{\AskInc@MainQuestion}
641 \AskInc@AtEnd%
642 \</package>

```

3 Test

3.1 Included files

```

643 <*a>
644 \chapter{Chapter A}
645 \let\filea=Y
646 </a>

647 <*b>
648 \chapter{Chapter B}
649 \let\fileb=Y
650 </b>

```

```

651 <*c>
652 \chapter{Chapter C}
653 \let\filec=Y
654 </c>

```

3.2 Special test setup

```

655 <*test18>
656 \includeonly{askinclude-b}
657 </test18>

658 <*test20 j test21 j test22>
659 \let\AtBeginDocumentMode=N%
660 </test20 j test21 j test22>

```

3.3 Common start

```

661 <*test0>
662 \ifx\CompatibilityMode Y%
663 \documentstyle[filecontents,qstest,askinclude]{book}%
664 \else
665 \documentclass{book}%
666 \usepackage{filecontents}%
667 \usepackage{qstest}%
668 <*test1>
669 \makeatletter
670 \def\RestoreCatcodes{%
671 \count@=0 %
672 \loop
673 \edef\RestoreCatcodes{%
674 \RestoreCatcodes
675 \catcode\the\count@=\the\catcode\count@\relax
676 }%
677 \ifnum\count@<255 %
678 \advance\count@\@ne
679 \repeat
680 \def\RangeCatcodeInvalid#1#2{%
681 \count@=#1\relax
682 \loop
683 \catcode\count@=15 %
684 \ifnum\count@<#2\relax
685 \advance\count@\@ne
686 \repeat
687 }%
688 \def\Test#1\EndTest{%
689 \RangeCatcodeInvalid{0}{47}%
690 \RangeCatcodeInvalid{58}{64}%
691 \RangeCatcodeInvalid{91}{96}%
692 \RangeCatcodeInvalid{123}{255}%
693 \catcode`\@=12 %
694 \catcode`\|=0 %
695 \catcode`\{=1 %
696 \catcode`\}=2 %
697 \catcode`\#=6 %
698 \catcode`\[=12 %
699 \catcode`\]=12 %
700 \catcode`\%=14 %
701 \catcode`\ =10 %
702 \catcode13=5 %
703 #1\relax
704 \RestoreCatcodes
705 \makeatother
706 }%
707 \Test
708 </test1>

```

```

709 <!*test1>
710   \ifx\AtBeginDocumentMode Y%
711     \AtBeginDocument{%
712       \usepackage{askinclude}[2011/12/02]%
713     }%
714   \else
715 </!*test1>
716     \usepackage{askinclude}[2011/12/02]%
717 <!*test1>
718   \fi
719 </!*test1>
720 <*test1>
721   \EndTest
722 </test1>
723 \fi
724 \IncludeTests{*}
725 \LogTests{log}{*}{*}
726 \makeatletter
727 \let\org@typein\typein
728 \def\typein[#1]#2{%
729   \ifx\@Stack\@StackEmpty
730     \begin{qstest}{StackNonEmpty}{StackNonEmpty}%
731       \Expect{Error: empty stack}{}%
732     \end{qstest}%
733     \org@typein[{#1}]{#2}%
734   \else
735     \edef#1{%
736       \unexpanded\expandafter\expandafter\expandafter{%
737         \expandafter\@car\@Stack\@nil
738       }%
739     }%
740     \xdef\@Stack{%
741       \unexpanded\expandafter\expandafter\expandafter{%
742         \expandafter\@cdr\@Stack\@nil
743       }%
744     }%
745     \typeout{%
746       \typeout{\string#1=#1}%
747     \fi
748 }
749 \def\@StackEmpty{\BOTTOM}
750 \let\@Stack\@StackEmpty
751 \newcommand*{\Stack}[1]{%
752   \def\@Stack{#1\BOTTOM}%
753 }
754 \AtEndDocument{%
755   \begin{qstest}{StackUsed}{StackUsed}%
756     \Expect*{%
757       \ifx\@Stack\@StackEmpty
758         \else
759           Error: Unused stack entries%
760         \fi
761       }{%
762         \end{qstest}%
763 }
764 \newcommand*{\CheckInclude}[2]{%
765   \include{askinclude-#1}%
766   \begin{qstest}{CheckInclude-#1}{CheckInclude-#1}%
767     \Expect*{%
768       \expandafter\ifx\csname file#1\endcsname Ytrue\else false\fi
769     }{#2}%
770   \end{qstest}%

```

```

771 }
772 \def\@EndChecks{}
773 \AtBeginDocument{%
774   \AtEndDocument{%
775     \if@filesw
776       \immediate\write\@mainaux{\string\@EndChecks}%
777     \fi
778   }%
779 }
780 \newcommand*\CheckAnswer[1]{%
781   \gdef\AnswerText{#1}%
782   \g@addto@macro\@EndChecks{\@CheckAnswer}%
783 }
784 \def\@CheckAnswer{%
785   \begin{qstest}{\CheckAnswer}{\CheckAnswer}%
786     \Expect*{\unexpanded\expandafter{\AskInc@Answer}}%
787     *{\unexpanded\expandafter{\AnswerText}}%
788   \end{qstest}%
789 }
790 \newcommand*\CheckMatchMethod[1]{%
791   \gdef\MatchMethodText{#1}%
792   \g@addto@macro\@EndChecks{\@CheckMatchMethod}%
793 }
794 \def\@CheckMatchMethod{%
795   \begin{qstest}{\CheckMatchMethod}{\CheckMatchMethod}%
796     \Expect*{\unexpanded\expandafter{\AskInc@MatchMethod}}%
797     *{\unexpanded\expandafter{\MatchMethodText}}%
798   \end{qstest}%
799 }
800 \def\@CheckAux{%
801   \begin{qstest}{\CheckAux}{\CheckAux}%
802     \def\GetFileContents##1{%
803       \pdfunescapehex{%
804         \pdffiledump length\pdffilesize{##1}{##1}%
805       }%
806     }%
807     \Expect*{\GetFileContents{\jobname.aux}}%
808     *{\GetFileContents{\jobname-aux.aux}}%
809   \end{qstest}%
810 }
811 \g@addto@macro\@EndChecks{\@CheckAux}
812 \let\org@filecontents\filecontents
813 \def\filecontents{%
814   \catcode\|=0 %
815   \org@filecontents
816 }
817 \begingroup
818 \endlinechar=-1 %
819 \catcode\|=0 %
820 \@makeother\#%
821 \@makeother\%
822 \gdef\AddList{%
823   \providecommand\AskInc@AddList[2]{%
824     \ifcase\ifx#1\@undefined
825       \else\ifx#1\relax
826       \else\ifx#1\@empty
827       \else1\fi\fi\fi0 %
828     \gdef#1{#2}%
829   \else
830     \g@addto@macro#1{,#2}%
831   \fi
832 }%

```

```

833 }%
834 |gdef|AddFile{%
835   \providecommand\AskInc@AddFile{%
836     \AskInc@AddList\AskInc@Files
837   }%
838 }%
839 |gdef|AddMatch{%
840   \providecommand\AskInc@AddMatch{%
841     \AskInc@AddList\AskInc@MatchFiles
842   }%
843 }%
844 |endgroup
845 </test0>

```

3.4 First run tests

```

846 <*test1>
847 \Stack{[]}
848 \CheckAnswer{*}
849 \begin{filecontents*}{\jobname.aux}
850 \relax
851 \end{filecontents*}
852 \begin{filecontents*}{\jobname-aux.aux}
853 \relax|space
854 |gdef\AskInc@Answer{*}
855 |gdef\AskInc@MatchMethod{}
856 |AddList
857 |AddFile
858 \AskInc@AddFile{askinclude-a}
859 \@input{askinclude-a.aux}
860 \AskInc@AddFile{askinclude-b}
861 \@input{askinclude-b.aux}
862 \AskInc@AddFile{askinclude-c}
863 \@input{askinclude-c.aux}
864 \@EndChecks
865 \end{filecontents*}
866 \begin{document}
867 \CheckInclude{a}{true}
868 \CheckInclude{b}{true}
869 \CheckInclude{c}{true}
870 </test1>

871 <*test2>
872 \Stack{*}
873 \CheckAnswer{*}
874 \begin{filecontents*}{\jobname.aux}
875 \relax
876 \end{filecontents*}
877 \begin{filecontents*}{\jobname-aux.aux}
878 \relax|space
879 |gdef\AskInc@Answer{*}
880 |gdef\AskInc@MatchMethod{}
881 |AddList
882 |AddFile
883 \AskInc@AddFile{askinclude-a}
884 \@input{askinclude-a.aux}
885 \AskInc@AddFile{askinclude-b}
886 \@input{askinclude-b.aux}
887 \AskInc@AddFile{askinclude-c}
888 \@input{askinclude-c.aux}
889 \@EndChecks
890 \end{filecontents*}
891 \begin{document}
892 \CheckInclude{a}{true}

```



```

893 \CheckInclude{b}{true}
894 \CheckInclude{c}{true}
895 </test2>
896 <*test3>
897 \Stack{-}%
898 \CheckAnswer{-}
899 \begin{filecontents*}{\jobname.aux}
900 \relax
901 \end{filecontents*}
902 \begin{filecontents*}{\jobname-aux.aux}
903 \relax|space
904 \gdef\AskInc@Answer{-}
905 \gdef\AskInc@MatchMethod{}
906 |AddList
907 |AddFile
908 \AskInc@AddFile{askinclude-a}
909 \@input{askinclude-a.aux}
910 \AskInc@AddFile{askinclude-b}
911 \@input{askinclude-b.aux}
912 \AskInc@AddFile{askinclude-c}
913 \@input{askinclude-c.aux}
914 \@EndChecks
915 \end{filecontents*}
916 \begin{document}
917 \CheckInclude{a}{false}
918 \CheckInclude{b}{false}
919 \CheckInclude{c}{false}
920 </test3>
921 <*test4>
922 \Stack{{askinclude-b}}
923 \CheckAnswer{askinclude-b}
924 \begin{filecontents*}{\jobname.aux}
925 \relax
926 \end{filecontents*}
927 \begin{filecontents*}{\jobname-aux.aux}
928 \relax|space
929 \gdef\AskInc@Answer{askinclude-b}
930 \gdef\AskInc@MatchMethod{}
931 |AddList
932 |AddFile
933 \AskInc@AddFile{askinclude-a}
934 \@input{askinclude-a.aux}
935 \AskInc@AddFile{askinclude-b}
936 \@input{askinclude-b.aux}
937 \AskInc@AddFile{askinclude-c}
938 \@input{askinclude-c.aux}
939 \@EndChecks
940 \end{filecontents*}
941 \begin{document}
942 \CheckInclude{a}{false}
943 \CheckInclude{b}{true}
944 \CheckInclude{c}{false}
945 </test4>
946 <*test5>
947 \Stack{{askinclude-a,askinclude-c}}
948 \CheckAnswer{askinclude-a,askinclude-c}
949 \begin{filecontents*}{\jobname.aux}
950 \relax
951 \end{filecontents*}
952 \begin{filecontents*}{\jobname-aux.aux}
953 \relax|space
954 \gdef\AskInc@Answer{askinclude-a,askinclude-c}

```

```

955 \gdef\AskInc@MatchMethod{}
956 |AddList
957 |AddFile
958 \AskInc@AddFile{askinclude-a}
959 \@input{askinclude-a.aux}
960 \AskInc@AddFile{askinclude-b}
961 \@input{askinclude-b.aux}
962 \AskInc@AddFile{askinclude-c}
963 \@input{askinclude-c.aux}
964 \@EndChecks
965 \end{filecontents*}
966 \begin{document}
967 \CheckInclude{a}{true}
968 \CheckInclude{b}{false}
969 \CheckInclude{c}{true}
970 </test5>

971 <*test6>
972 \Stack{?yny}
973 \CheckAnswer{askinclude-a,askinclude-c}
974 \begin{filecontents*}{\jobname.aux}
975 \relax
976 \end{filecontents*}
977 \begin{filecontents*}{\jobname-aux.aux}
978 \relax|space
979 \gdef\AskInc@Answer{}
980 \gdef\AskInc@MatchMethod{}
981 |AddList
982 |AddFile
983 \AskInc@AddFile{askinclude-a}
984 \gdef\AskInc@Answer{askinclude-a}
985 \@input{askinclude-a.aux}
986 \AskInc@AddFile{askinclude-b}
987 \@input{askinclude-b.aux}
988 \AskInc@AddFile{askinclude-c}
989 \gdef\AskInc@Answer{askinclude-a,askinclude-c}
990 \@input{askinclude-c.aux}
991 \@EndChecks
992 \end{filecontents*}
993 \begin{document}
994 \CheckInclude{a}{true}
995 \CheckInclude{b}{false}
996 \CheckInclude{c}{true}
997 </test6>

998 <*test7>
999 \Stack{?yA}
1000 \CheckAnswer{askinclude-a,askinclude-b,askinclude-c}
1001 \begin{filecontents*}{\jobname.aux}
1002 \relax
1003 \end{filecontents*}
1004 \begin{filecontents*}{\jobname-aux.aux}
1005 \relax|space
1006 \gdef\AskInc@Answer{}
1007 \gdef\AskInc@MatchMethod{}
1008 |AddList
1009 |AddFile
1010 \AskInc@AddFile{askinclude-a}
1011 \gdef\AskInc@Answer{askinclude-a}
1012 \@input{askinclude-a.aux}
1013 \AskInc@AddFile{askinclude-b}
1014 \gdef\AskInc@Answer{askinclude-a,askinclude-b}
1015 \@input{askinclude-b.aux}
1016 \AskInc@AddFile{askinclude-c}

```

```

1017 \gdef\AskInc@Answer{askinclude-a,askinclude-b,askinclude-c}
1018 \@input{askinclude-c.aux}
1019 \@EndChecks
1020 \end{filecontents*}
1021 \begin{document}
1022 \CheckInclude{a}{true}
1023 \CheckInclude{b}{true}
1024 \CheckInclude{c}{true}
1025 \end{test7}

1026 \begin{test8}
1027 \Stack{?yD}
1028 \CheckAnswer{askinclude-a,askinclude-b,askinclude-c}
1029 \begin{filecontents*}{\jobname.aux}
1030 \relax
1031 \end{filecontents*}
1032 \begin{filecontents*}{\jobname-aux.aux}
1033 \relax\space
1034 \gdef\AskInc@Answer{}
1035 \gdef\AskInc@MatchMethod{}
1036 |AddList
1037 |AddFile
1038 \AskInc@AddFile{askinclude-a}
1039 \gdef\AskInc@Answer{askinclude-a}
1040 \@input{askinclude-a.aux}
1041 \AskInc@AddFile{askinclude-b}
1042 \gdef\AskInc@Answer{askinclude-a,askinclude-b}
1043 \@input{askinclude-b.aux}
1044 \AskInc@AddFile{askinclude-c}
1045 \gdef\AskInc@Answer{askinclude-a,askinclude-b,askinclude-c}
1046 \@input{askinclude-c.aux}
1047 \@EndChecks
1048 \end{filecontents*}
1049 \begin{document}
1050 \CheckInclude{a}{true}
1051 \CheckInclude{b}{true}
1052 \CheckInclude{c}{true}
1053 \end{test8}

1054 \begin{test9}
1055 \Stack{?yN}
1056 \CheckAnswer{askinclude-a}
1057 \begin{filecontents*}{\jobname.aux}
1058 \relax
1059 \end{filecontents*}
1060 \begin{filecontents*}{\jobname-aux.aux}
1061 \relax\space
1062 \gdef\AskInc@Answer{}
1063 \gdef\AskInc@MatchMethod{}
1064 |AddList
1065 |AddFile
1066 \AskInc@AddFile{askinclude-a}
1067 \gdef\AskInc@Answer{askinclude-a}
1068 \@input{askinclude-a.aux}
1069 \AskInc@AddFile{askinclude-b}
1070 \@input{askinclude-b.aux}
1071 \AskInc@AddFile{askinclude-c}
1072 \@input{askinclude-c.aux}
1073 \@EndChecks
1074 \end{filecontents*}
1075 \begin{document}
1076 \CheckInclude{a}{true}
1077 \CheckInclude{b}{false}
1078 \CheckInclude{c}{false}

```

```

1079 </test9>
1080 <*test10>
1081 \Stack{?{}N}
1082 \CheckAnswer{askinclude-a}
1083 \begin{filecontents*}{\jobname.aux}
1084 \relax
1085 \end{filecontents*}
1086 \begin{filecontents*}{\jobname-aux.aux}
1087 \relax\space
1088 \gdef\AskInc@Answer{}
1089 \gdef\AskInc@MatchMethod{}
1090 |AddList
1091 |AddFile
1092 \AskInc@AddFile{askinclude-a}
1093 \gdef\AskInc@Answer{askinclude-a}
1094 \@input{askinclude-a.aux}
1095 \AskInc@AddFile{askinclude-b}
1096 \@input{askinclude-b.aux}
1097 \AskInc@AddFile{askinclude-c}
1098 \@input{askinclude-c.aux}
1099 \@EndChecks
1100 \end{filecontents*}
1101 \begin{document}
1102 \CheckInclude{a}{true}
1103 \CheckInclude{b}{false}
1104 \CheckInclude{c}{false}
1105 </test10>
1106 <*test11>
1107 \Stack{?xxyxxN}
1108 \CheckAnswer{askinclude-a}
1109 \begin{filecontents*}{\jobname.aux}
1110 \relax
1111 \end{filecontents*}
1112 \begin{filecontents*}{\jobname-aux.aux}
1113 \relax\space
1114 \gdef\AskInc@Answer{}
1115 \gdef\AskInc@MatchMethod{}
1116 |AddList
1117 |AddFile
1118 \AskInc@AddFile{askinclude-a}
1119 \gdef\AskInc@Answer{askinclude-a}
1120 \@input{askinclude-a.aux}
1121 \AskInc@AddFile{askinclude-b}
1122 \@input{askinclude-b.aux}
1123 \AskInc@AddFile{askinclude-c}
1124 \@input{askinclude-c.aux}
1125 \@EndChecks
1126 \end{filecontents*}
1127 \begin{document}
1128 \CheckInclude{a}{true}
1129 \CheckInclude{b}{false}
1130 \CheckInclude{c}{false}
1131 </test11>

```

3.5 Second run tests

```

1132 <*test12>
1133 \Stack{{}}
1134 \CheckAnswer{askinclude-b}
1135 \begin{filecontents*}{\jobname.aux}
1136 \gdef\AskInc@Answer{askinclude-b}
1137 \gdef\AskInc@MatchMethod{}
1138 |AddList

```

```

1139 |AddFile
1140 \AskInc@AddFile{askinclude-a}
1141 \AskInc@AddFile{askinclude-b}
1142 \AskInc@AddFile{askinclude-c}
1143 \end{filecontents*}
1144 \begin{filecontents*}{\jobname-aux.aux}
1145 \relax|space
1146 \gdef\AskInc@Answer{askinclude-b}
1147 \gdef\AskInc@MatchMethod{}
1148 |AddList
1149 |AddFile
1150 \AskInc@AddFile{askinclude-a}
1151 \@input{askinclude-a.aux}
1152 \AskInc@AddFile{askinclude-b}
1153 \@input{askinclude-b.aux}
1154 \AskInc@AddFile{askinclude-c}
1155 \@input{askinclude-c.aux}
1156 \@EndChecks
1157 \end{filecontents*}
1158 \begin{document}
1159 \CheckInclude{a}{false}
1160 \CheckInclude{b}{true}
1161 \CheckInclude{c}{false}
1162 </test12>

1163 <*test13>
1164 \Stack{{}}
1165 \CheckAnswer{askinclude-b}
1166 \begin{filecontents*}{\jobname.aux}
1167 \gdef\AskInc@Answer{askinclude-b}
1168 \gdef\AskInc@MatchMethod{}
1169 |AddList
1170 |AddFile
1171 \AskInc@AddFile{askinclude-a}
1172 \AskInc@AddFile{askinclude-b}
1173 \end{filecontents*}
1174 \begin{filecontents*}{\jobname-aux.aux}
1175 \relax|space
1176 \gdef\AskInc@Answer{askinclude-b}
1177 \gdef\AskInc@MatchMethod{}
1178 |AddList
1179 |AddFile
1180 \AskInc@AddFile{askinclude-a}
1181 \@input{askinclude-a.aux}
1182 \AskInc@AddFile{askinclude-b}
1183 \@input{askinclude-b.aux}
1184 \AskInc@AddFile{askinclude-c}
1185 \@input{askinclude-c.aux}
1186 \@EndChecks
1187 \end{filecontents*}
1188 \begin{document}
1189 \CheckInclude{a}{false}
1190 \CheckInclude{b}{true}
1191 \CheckInclude{c}{false}
1192 </test13>

1193 <*test14>
1194 \Stack{{}}
1195 \CheckAnswer{*}
1196 \begin{filecontents*}{\jobname.aux}
1197 \gdef\AskInc@Answer{*}
1198 \gdef\AskInc@MatchMethod{}
1199 |AddList
1200 |AddFile

```

```

1201 \AskInc@AddFile{askinclude-a}
1202 \AskInc@AddFile{askinclude-b}
1203 \end{filecontents*}
1204 \begin{filecontents*}{\jobname-aux.aux}
1205 \relax\space
1206 \gdef\AskInc@Answer{*}
1207 \gdef\AskInc@MatchMethod{}
1208 |AddList
1209 |AddFile
1210 \AskInc@AddFile{askinclude-a}
1211 \@input{askinclude-a.aux}
1212 \AskInc@AddFile{askinclude-b}
1213 \@input{askinclude-b.aux}
1214 \AskInc@AddFile{askinclude-c}
1215 \@input{askinclude-c.aux}
1216 \@EndChecks
1217 \end{filecontents*}
1218 \begin{document}
1219 \CheckInclude{a}{true}
1220 \CheckInclude{b}{true}
1221 \CheckInclude{c}{true}
1222 </test14>

1223 <*test15>
1224 \Stack{?D}
1225 \CheckAnswer{askinclude-a,askinclude-c}
1226 \begin{filecontents*}{\jobname.aux}
1227 \gdef\AskInc@Answer{askinclude-a}
1228 \gdef\AskInc@MatchMethod{}
1229 |AddList
1230 |AddFile
1231 \AskInc@AddFile{askinclude-a}
1232 \AskInc@AddFile{askinclude-b}
1233 \end{filecontents*}
1234 \begin{filecontents*}{\jobname-aux.aux}
1235 \relax\space
1236 \gdef\AskInc@Answer{}
1237 \gdef\AskInc@MatchMethod{}
1238 |AddList
1239 |AddFile
1240 \AskInc@AddFile{askinclude-a}
1241 \gdef\AskInc@Answer{askinclude-a}
1242 \@input{askinclude-a.aux}
1243 \AskInc@AddFile{askinclude-b}
1244 \@input{askinclude-b.aux}
1245 \AskInc@AddFile{askinclude-c}
1246 \gdef\AskInc@Answer{askinclude-a,askinclude-c}
1247 \@input{askinclude-c.aux}
1248 \@EndChecks
1249 \end{filecontents*}
1250 \begin{document}
1251 \CheckInclude{a}{true}
1252 \CheckInclude{b}{false}
1253 \CheckInclude{c}{true}
1254 </test15>

1255 <*test16>
1256 \Stack{?A}
1257 \CheckAnswer{askinclude-a,askinclude-b,askinclude-c}
1258 \begin{filecontents*}{\jobname.aux}
1259 \gdef\AskInc@Answer{askinclude-a}
1260 \gdef\AskInc@MatchMethod{}
1261 |AddList
1262 |AddFile

```

```

1263 \AskInc@AddFile{askinclude-a}
1264 \AskInc@AddFile{askinclude-b}
1265 \end{filecontents*}
1266 \begin{filecontents*}{\jobname-aux.aux}
1267 \relax\space
1268 \gdef\AskInc@Answer{}
1269 \gdef\AskInc@MatchMethod{}
1270 |AddList
1271 |AddFile
1272 \AskInc@AddFile{askinclude-a}
1273 \gdef\AskInc@Answer{askinclude-a}
1274 \@input{askinclude-a.aux}
1275 \AskInc@AddFile{askinclude-b}
1276 \gdef\AskInc@Answer{askinclude-a,askinclude-b}
1277 \@input{askinclude-b.aux}
1278 \AskInc@AddFile{askinclude-c}
1279 \gdef\AskInc@Answer{askinclude-a,askinclude-b,askinclude-c}
1280 \@input{askinclude-c.aux}
1281 \@EndChecks
1282 \end{filecontents*}
1283 \begin{document}
1284 \CheckInclude{a}{true}
1285 \CheckInclude{b}{true}
1286 \CheckInclude{c}{true}
1287 </test16>

1288 <*test17>
1289 \Stack{?N}
1290 \CheckAnswer{}
1291 \begin{filecontents*}{\jobname.aux}
1292 \gdef\AskInc@Answer{askinclude-a}
1293 \gdef\AskInc@MatchMethod{}
1294 |AddList
1295 |AddFile
1296 \AskInc@AddFile{askinclude-a}
1297 \AskInc@AddFile{askinclude-b}
1298 \end{filecontents*}
1299 \begin{filecontents*}{\jobname-aux.aux}
1300 \relax\space
1301 \gdef\AskInc@Answer{}
1302 \gdef\AskInc@MatchMethod{}
1303 |AddList
1304 |AddFile
1305 \AskInc@AddFile{askinclude-a}
1306 \@input{askinclude-a.aux}
1307 \AskInc@AddFile{askinclude-b}
1308 \@input{askinclude-b.aux}
1309 \AskInc@AddFile{askinclude-c}
1310 \@input{askinclude-c.aux}
1311 \@EndChecks
1312 \end{filecontents*}
1313 \begin{document}
1314 \CheckInclude{a}{false}
1315 \CheckInclude{b}{false}
1316 \CheckInclude{c}{false}
1317 </test17>

1318 <*test18>
1319 \Stack{}
1320 \CheckAnswer{askinclude-a}
1321 \begin{filecontents*}{\jobname.aux}
1322 \gdef\AskInc@Answer{askinclude-a}
1323 \gdef\AskInc@MatchMethod{}
1324 |AddList

```

```

1325 |AddFile
1326 \AskInc@AddFile{askinclude-a}
1327 \AskInc@AddFile{askinclude-b}
1328 \end{filecontents*}
1329 \begin{filecontents*}{\jobname-aux.aux}
1330 \relax|space
1331 \gdef\AskInc@Answer{askinclude-a}
1332 \gdef\AskInc@MatchMethod{}
1333 |AddList
1334 |AddFile
1335 \AskInc@AddFile{askinclude-a}
1336 \@input{askinclude-a.aux}
1337 \AskInc@AddFile{askinclude-b}
1338 \@input{askinclude-b.aux}
1339 \AskInc@AddFile{askinclude-c}
1340 \@input{askinclude-c.aux}
1341 \@EndChecks
1342 \end{filecontents*}
1343 \begin{document}
1344 \CheckInclude{a}{false}
1345 \CheckInclude{b}{true}
1346 \CheckInclude{c}{false}
1347 </test18>

1348 <*test19>
1349 \Stack{?N}
1350 \CheckAnswer{}
1351 \begin{filecontents*}{\jobname.aux}
1352 \gdef\AskInc@Answer{askinclude-a}
1353 \gdef\AskInc@MatchMethod{}
1354 |AddList
1355 |AddFile
1356 \AskInc@AddFile{askinclude-a}
1357 \AskInc@AddFile{askinclude-b}
1358 \end{filecontents*}
1359 \begin{filecontents*}{\jobname-aux.aux}
1360 \relax|space
1361 \gdef\AskInc@Answer{}
1362 \gdef\AskInc@MatchMethod{}
1363 |AddList
1364 |AddFile
1365 \AskInc@AddFile{askinclude-a}
1366 \@input{askinclude-a.aux}
1367 \AskInc@AddFile{askinclude-b}
1368 \@input{askinclude-b.aux}
1369 \AskInc@AddFile{askinclude-c}
1370 \@input{askinclude-c.aux}
1371 \@EndChecks
1372 \end{filecontents*}
1373 \ifx\AtBeginDocumentMode Y%
1374 \AtBeginDocument{\includeonly{askinclude-b}}%
1375 \else
1376 \includeonly{askinclude-b}%
1377 \fi
1378 \begin{document}
1379 \CheckInclude{a}{false}
1380 \CheckInclude{b}{false}
1381 \CheckInclude{c}{false}
1382 </test19>

```

3.6 Compatibility for changing from old to new

```

1383 <*test20>
1384 \Stack{{}}

```



```

1385 \CheckAnswer{*}
1386 \begin{filecontents*}{\jobname.aux}
1387 \@includeonly{*}
1388 \end{filecontents*}
1389 \begin{filecontents*}{\jobname-aux.aux}
1390 \relax\space
1391 \gdef\AskInc@Answer{*}
1392 \gdef\AskInc@MatchMethod{}
1393 |AddList
1394 |AddFile
1395 \AskInc@AddFile{askinclude-a}
1396 \@input{askinclude-a.aux}
1397 \AskInc@AddFile{askinclude-b}
1398 \@input{askinclude-b.aux}
1399 \AskInc@AddFile{askinclude-c}
1400 \@input{askinclude-c.aux}
1401 \@EndChecks
1402 \end{filecontents*}
1403 \begin{document}
1404 \CheckInclude{a}{true}
1405 \CheckInclude{b}{true}
1406 \CheckInclude{c}{true}
1407 </test20>

1408 <*test21>
1409 \Stack{[]}
1410 \CheckAnswer{-}
1411 \begin{filecontents*}{\jobname.aux}
1412 \@includeonly{}
1413 \end{filecontents*}
1414 \begin{filecontents*}{\jobname-aux.aux}
1415 \relax\space
1416 \gdef\AskInc@Answer{-}
1417 \gdef\AskInc@MatchMethod{}
1418 |AddList
1419 |AddFile
1420 \AskInc@AddFile{askinclude-a}
1421 \@input{askinclude-a.aux}
1422 \AskInc@AddFile{askinclude-b}
1423 \@input{askinclude-b.aux}
1424 \AskInc@AddFile{askinclude-c}
1425 \@input{askinclude-c.aux}
1426 \@EndChecks
1427 \end{filecontents*}
1428 \begin{document}
1429 \CheckInclude{a}{false}
1430 \CheckInclude{b}{false}
1431 \CheckInclude{c}{false}
1432 </test21>

1433 <*test22>
1434 \Stack{[]}
1435 \CheckAnswer{askinclude-a,askinclude-c}
1436 \begin{filecontents*}{\jobname.aux}
1437 \@includeonly{askinclude-a,askinclude-c}
1438 \end{filecontents*}
1439 \begin{filecontents*}{\jobname-aux.aux}
1440 \relax\space
1441 \gdef\AskInc@Answer{askinclude-a,askinclude-c}
1442 \gdef\AskInc@MatchMethod{}
1443 |AddList
1444 |AddFile
1445 \AskInc@AddFile{askinclude-a}
1446 \@input{askinclude-a.aux}

```

```

1447 \AskInc@AddFile{askinclude-b}
1448 \@input{askinclude-b.aux}
1449 \AskInc@AddFile{askinclude-c}
1450 \@input{askinclude-c.aux}
1451 \@EndChecks
1452 \end{filecontents*}
1453 \begin{document}
1454 \CheckInclude{a}{true}
1455 \CheckInclude{b}{false}
1456 \CheckInclude{c}{true}
1457 </test22>

```

3.7 Regular expressions

```

1458 <*test23>
1459 \Stack{{\makematch}{*,!askinclude-b}}
1460 \CheckAnswer{*,!askinclude-b}
1461 \CheckMatchMethod{\makematch}
1462 \begin{filecontents*}{\jobname.aux}
1463 \relax
1464 \end{filecontents*}
1465 \begin{filecontents*}{\jobname-aux.aux}
1466 \relax|space
1467 \gdef\AskInc@Answer{*,!askinclude-b}
1468 \gdef\AskInc@MatchMethod{\makematch}
1469 |AddList
1470 |AddFile
1471 \AskInc@AddFile{askinclude-a}
1472 |AddMatch
1473 \AskInc@AddMatch{askinclude-a}
1474 \@input{askinclude-a.aux}
1475 \AskInc@AddFile{askinclude-b}
1476 \@input{askinclude-b.aux}
1477 \AskInc@AddFile{askinclude-c}
1478 \AskInc@AddMatch{askinclude-c}
1479 \@input{askinclude-c.aux}
1480 \@EndChecks
1481 \end{filecontents*}
1482 \begin{document}
1483 \CheckInclude{a}{true}
1484 \CheckInclude{b}{false}
1485 \CheckInclude{c}{true}
1486 </test23>

1487 <*test24>
1488 \Stack{{\pdfmatch}{.,!askinclude-b}}
1489 \CheckAnswer{.,!askinclude-b}
1490 \CheckMatchMethod{\pdfmatch}
1491 \begin{filecontents*}{\jobname.aux}
1492 \relax
1493 \end{filecontents*}
1494 \begin{filecontents*}{\jobname-aux.aux}
1495 \relax|space
1496 \gdef\AskInc@Answer{.,!askinclude-b}
1497 \gdef\AskInc@MatchMethod{\pdfmatch}
1498 |AddList
1499 |AddFile
1500 \AskInc@AddFile{askinclude-a}
1501 |AddMatch
1502 \AskInc@AddMatch{askinclude-a}
1503 \@input{askinclude-a.aux}
1504 \AskInc@AddFile{askinclude-b}
1505 \@input{askinclude-b.aux}
1506 \AskInc@AddFile{askinclude-c}

```

```

1507 \AskInc@AddMatch{askinclude-c}
1508 \@input{askinclude-c.aux}
1509 \@EndChecks
1510 \end{filecontents*}
1511 \begin{document}
1512 \CheckInclude{a}{true}
1513 \CheckInclude{b}{false}
1514 \CheckInclude{c}{true}
1515 \</test24>

```

3.8 Common end

```

1516 \<*test0>
1517 \end{document}
1518 \</test0>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/askinclude.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/askinclude.pdf](#) Documentation.

Bundle. All the packages of the bundle ‘oberdiek’ 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/oberdiek.tds.zip](#)

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

4.2 Bundle installation

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

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

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```

chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/

```

4.3 Package installation

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

```
tex askinclude.dtx
```

¹[ftp://ftp.ctan.org/tex-archive/](http://ftp.ctan.org/tex-archive/)

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

```

askinclude.sty      → tex/latex/oberdiek/askinclude.sty
askinclude.pdf      → doc/latex/oberdiek/askinclude.pdf
test/askinclude-a.tex → doc/latex/oberdiek/test/askinclude-a.tex
test/askinclude-b.tex → doc/latex/oberdiek/test/askinclude-b.tex
test/askinclude-c.tex → doc/latex/oberdiek/test/askinclude-c.tex
test/askinclude-test1.tex → doc/latex/oberdiek/test/askinclude-test1.tex
test/askinclude-test2.tex → doc/latex/oberdiek/test/askinclude-test2.tex
test/askinclude-test3.tex → doc/latex/oberdiek/test/askinclude-test3.tex
test/askinclude-test4.tex → doc/latex/oberdiek/test/askinclude-test4.tex
test/askinclude-test5.tex → doc/latex/oberdiek/test/askinclude-test5.tex
test/askinclude-test6.tex → doc/latex/oberdiek/test/askinclude-test6.tex
test/askinclude-test7.tex → doc/latex/oberdiek/test/askinclude-test7.tex
test/askinclude-test8.tex → doc/latex/oberdiek/test/askinclude-test8.tex
test/askinclude-test9.tex → doc/latex/oberdiek/test/askinclude-test9.tex
test/askinclude-test10.tex → doc/latex/oberdiek/test/askinclude-test10.tex
test/askinclude-test11.tex → doc/latex/oberdiek/test/askinclude-test11.tex
test/askinclude-test12.tex → doc/latex/oberdiek/test/askinclude-test12.tex
test/askinclude-test13.tex → doc/latex/oberdiek/test/askinclude-test13.tex
test/askinclude-test14.tex → doc/latex/oberdiek/test/askinclude-test14.tex
test/askinclude-test15.tex → doc/latex/oberdiek/test/askinclude-test15.tex
test/askinclude-test16.tex → doc/latex/oberdiek/test/askinclude-test16.tex
test/askinclude-test17.tex → doc/latex/oberdiek/test/askinclude-test17.tex
test/askinclude-test18.tex → doc/latex/oberdiek/test/askinclude-test18.tex
test/askinclude-test19.tex → doc/latex/oberdiek/test/askinclude-test19.tex
test/askinclude-test20.tex → doc/latex/oberdiek/test/askinclude-test20.tex
test/askinclude-test21.tex → doc/latex/oberdiek/test/askinclude-test21.tex
test/askinclude-test22.tex → doc/latex/oberdiek/test/askinclude-test22.tex
test/askinclude-test23.tex → doc/latex/oberdiek/test/askinclude-test23.tex
test/askinclude-test24.tex → doc/latex/oberdiek/test/askinclude-test24.tex
askinclude.dtx      → source/latex/oberdiek/askinclude.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`.

4.4 Refresh file name databases

If your \TeX distribution (`te \TeX` , `mik \TeX` , ...) relies on file name databases, you must refresh these. For example, `te \TeX` users run `texhash` or `mktextlsr`.

4.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk askinclude.pdf unpack_files output .
```

Unpacking with \LaTeX . The `.dtx` chooses its action depending on the format:

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

\LaTeX : Generate the documentation.

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

```
latex \let\install=y\input{askinclude.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 askinclude.dtx
bibtex askinclude.aux
makeindex -s gind.ist askinclude.idx
pdflatex askinclude.dtx
makeindex -s gind.ist askinclude.idx
pdflatex askinclude.dtx
```

5 Catalogue

The following XML file can be used as source for the [T_EX Catalogue](#). The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `askinclude.xml`.

```
1519 <*catalogue>
1520 <?xml version='1.0' encoding='us-ascii'?>
1521 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
1522 <entry datestamp='$Date$' modifier='$Author$' id='askinclude'>
1523   <name>askinclude</name>
1524   <caption>Interactive use of \includeonly</caption>
1525   <authorref id='auth:straub'/>
1526   <authorref id='auth:oberdiek'/>
1527   <copyright owner="Pablo A. Straub" year='1991-1994'/>
1528   <copyright owner="Heiko Oberdiek" year='2007,2008,2011'/>
1529   <license type='lppl1.3'/>
1530   <version number='2.2'/>
1531   <description>
1532     The package asks the user which files to put in a \includeonly
1533     command. There is provision for answering &#x201C;same as last
1534     time&#x201D; or &#x201C;all files&#x201D;..
1535     <p/>
1536     The package is part of the <xref refid='oberdiek'>oberdiek</xref> bundle.
1537   </description>
1538   <documentation details='Package documentation'
1539     href='ctan:/macros/latex/contrib/oberdiek/askinclude.pdf'/>
1540   <ctan file='true' path='/macros/latex/contrib/oberdiek/askinclude.dtx'/>
1541   <miktex location='oberdiek'/>
1542   <texlive location='oberdiek'/>
1543   <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
1544 </entry>
1545 </catalogue>
```

6 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`.

7 References

- [1] Robin Fairbairns. *[tex-live] licence on includeonly.sty*. 2007-10-13. URL: <http://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: <http://mirror.ctan.org/macros/latex/contrib/excludeonly/excludeonly.pdf> (visited on 2011-12-02).
- [3] Herbert Voß. *askinclude.sty*. Thread in newsgroup `comp.text.tex`; Message-ID: `fev72m$b35$02$2@news.t-online.com`. 2007-10-15. URL: http://groups.google.com/group/comp.text.tex/browse_thread/thread/728a3e73c04c6789/349e31865c40d00c (visited on 2011-12-02).

8 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 LaTeX”²

[1991/12/13 v1.1]

- *Unknown*.

[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.

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

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

[2008/08/11 v2.1]

- Code is not changed.
- URLs updated.

[2011/12/02 v2.2]

- Bibliography rewritten for biblatex.

9 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 | |
|-------------------------|---|
| \! | 1040, 1043, 1046, 1068, 1070, 1072, 1094, 1096, 1098, 1120, |
| \# | 1122, 1124, 1151, 1153, 1155, |
| \\$ | 1181, 1183, 1185, 1211, 1213, |
| \% | 1215, 1242, 1244, 1247, 1274, |
| \(| 1277, 1280, 1306, 1308, 1310, |
| \) | 1336, 1338, 1340, 1366, 1368, |
| * | 1370, 1396, 1398, 1400, 1421, |
| \+ | 1423, 1425, 1446, 1448, 1450, |
| \, | 1474, 1476, 1479, 1503, 1505, 1508 |
| \- | \@mainaux 61, 776 |
| \. | \@makeother 351, 352, 353, 354, 355, |
| \: | 356, 357, 358, 359, 360, 361, |
| \? | 362, 363, 364, 365, 366, 820, 821 |
| \@ | \@ne 678, 685 |
| \@CheckAnswer | \@nil 473, 475, 737, 742 |
| \@CheckAux | \@partlist 435, 436, 438, 588 |
| \@CheckMatchMethod | \@secondoftwo 276, 560 |
| \@EndChecks | \@spaces 287, |
| | 312, 332, 338, 340, 341, 342, 368 |
| | \@undefined 70, 447, 458, 517, 824 |
| | \[..... 211, 361, 698 |
| | \] 210, 363, 694, 821 |
| | \{ 221, 695 |
| | \} 222, 696 |
| \@Stack | \] 212, 362, 699 |
| \@StackEmpty | \^ 219, 356 |
| \@car | \ 218, 364, 814, 819 |
| \@cdr | |
| \@empty | |
| | 136, 139, 152, 195, 226, 238, |
| | 247, 251, 262, 263, 273, 289, |
| | 305, 317, 374, 387, 390, 401, |
| | 402, 408, 411, 415, 435, 473, |
| | 500, 508, 547, 557, 582, 602, 826 |
| \@firstofone | 55, 456, 467 |
| \@firstoftwo | 274, 558 |
| \@for | 266, 278, 279, 562, 563, 572 |
| \@gobble | .. 101, 112, 113, 456, 469, 530 |
| \@ifundefined | 122, 127, |
| | 150, 231, 237, 241, 250, 543, 546 |
| \@includeonly | ... <u>134</u> , 1387, 1412, 1437 |
| \@input | 859, 861, 863, |
| | 884, 886, 888, 909, 911, 913, |
| | 934, 936, 938, 959, 961, 963, |
| | 985, 987, 990, 1012, 1015, 1018, |
| | 1040, 1043, 1046, 1068, 1070, |
| | 1072, 1094, 1096, 1098, 1120, |
| | 1122, 1124, 1151, 1153, 1155, |
| | 1181, 1183, 1185, 1211, 1213, |
| | 1215, 1242, 1244, 1247, 1274, |
| | 1277, 1280, 1306, 1308, 1310, |
| | 1336, 1338, 1340, 1366, 1368, |
| | 1370, 1396, 1398, 1400, 1421, |
| | 1423, 1425, 1446, 1448, 1450, |
| | 1474, 1476, 1479, 1503, 1505, 1508 |
| | \@mainaux 61, 776 |
| | \@makeother 351, 352, 353, 354, 355, |
| | 356, 357, 358, 359, 360, 361, |
| | 362, 363, 364, 365, 366, 820, 821 |
| | \@ne 678, 685 |
| | \@nil 473, 475, 737, 742 |
| | \@partlist 435, 436, 438, 588 |
| | \@secondoftwo 276, 560 |
| | \@spaces 287, |
| | 312, 332, 338, 340, 341, 342, 368 |
| | \@undefined 70, 447, 458, 517, 824 |
| | \[..... 211, 361, 698 |
| | \] 210, 363, 694, 821 |
| | \{ 221, 695 |
| | \} 222, 696 |
| | \] 212, 362, 699 |
| | \^ 219, 356 |
| | \ 218, 364, 814, 819 |
| | |
| | _ 701 |
| A | |
| \advance | 678, 685 |
| \answer | ... 367, 370, 592, 595, 600, 601 |
| \AnswerText | 781, 787 |
| \AskInc@@Answer | 370, 373, 376, |
| | 379, 390, 391, 400, 406, 407, 410 |
| \AskInc@@CheckMatch | 473, 475 |
| \AskInc@@QuestionHeader | 579, 593, 594 |
| \AskInc@AddFile | . <u>93</u> , <u>100</u> , 106, 835, |
| | 858, 860, 862, 883, 885, 887, |
| | 908, 910, 912, 933, 935, 937, |
| | 958, 960, 962, 983, 986, 988, |
| | 1010, 1013, 1016, 1038, 1041, |
| | 1044, 1066, 1069, 1071, 1092, |

1095, 1097, 1118, 1121, 1123,
 1140, 1141, 1142, 1150, 1152,
 1154, 1171, 1172, 1180, 1182,
 1184, 1201, 1202, 1210, 1212,
 1214, 1231, 1232, 1240, 1243,
 1245, 1263, 1264, 1272, 1275,
 1278, 1296, 1297, 1305, 1307,
 1309, 1326, 1327, 1335, 1337,
 1339, 1356, 1357, 1365, 1367,
 1369, 1395, 1397, 1399, 1420,
 1422, 1424, 1445, 1447, 1449,
 1471, 1475, 1477, 1500, 1504, 1506
 \AskInc@AddList
 69, 94, 523, 823, 836, 841
 \AskInc@AddMatch 522, 529,
 535, 840, 1473, 1478, 1502, 1507
 \AskInc@Answer
 124, 135, 136, 137, 232,
 234, 236, 267, 271, 278, 312,
 391, 401, 406, 417, 430, 540,
 582, 583, 585, 588, 786, 854,
 879, 904, 929, 954, 979, 984,
 989, 1006, 1011, 1014, 1017,
 1034, 1039, 1042, 1045, 1062,
 1067, 1088, 1093, 1114, 1119,
 1136, 1146, 1167, 1176, 1197,
 1206, 1227, 1236, 1241, 1246,
 1259, 1268, 1273, 1276, 1279,
 1292, 1301, 1322, 1331, 1352,
 1361, 1391, 1416, 1441, 1467, 1496
 \AskInc@AskFile 112, 116, 403
 \AskInc@AskQuestion 580, 591
 \AskInc@AskResult
 581, 600, 602, 603, 606, 609,
 612, 614, 617, 619, 622, 624, 628
 \AskInc@AtBeginDocument 54, 59
 \AskInc@AtEnd 29, 30, 51, 159, 641
 \AskInc@CharAll 205, 612
 \AskInc@CharAsterisk
 198, 232, 267, 407, 544, 551
 \AskInc@CharDefault 207, 622
 \AskInc@CharExclam 202, 477
 \AskInc@CharHash 66, 70, 72, 74, 81, 83
 \AskInc@CharMinus
 199, 270, 271, 410, 555
 \AskInc@CharNo
 204, 554, 570, 574, 609, 619
 \AskInc@CharNone 206, 617
 \AskInc@CharPlus .. 200, 268, 282, 288
 \AskInc@CharQuestion 201, 400
 \AskInc@CharYes
 203, 552, 566, 571, 581, 606, 614
 \AskInc@CheckMatch 465, 472
 \AskInc@Default 552, 554,
 566, 570, 571, 574, 597, 603, 624
 \AskInc@DoAskFile 403, 549
 \AskInc@DoInclude 426
 \AskInc@DoMatch 433, 451, 462
 \AskInc@File 266, 281,
 287, 290, 294, 550, 565, 573, 596
 \AskInc@Files 94, 109, 263, 266, 572, 836
 \AskInc@FilterPattern 502, 504
 \AskInc@Foundfalse 463, 480
 \AskInc@Foundtrue 487
 \AskInc@Include 426
 \AskInc@InvalidPattern . 482, 489, 493
 \AskInc@MainAnswer 393, 399
 \AskInc@MainQuestion 344, 640
 \AskInc@MakeMatchInit 443, 449
 \AskInc@Mark .. 268, 270, 282, 287, 288
 \AskInc@MatchFile 113, 117, 432
 \AskInc@MatchFiles
 155, 251, 279, 523, 563, 841
 \AskInc@MatchInit 420, 428
 \AskInc@MatchList 431,
 455, 465, 499, 502, 508, 509, 511
 \AskInc@MatchMethod 129, 139, 152,
 238, 240, 242, 273, 305, 308,
 402, 408, 411, 414, 442, 796,
 855, 880, 905, 930, 955, 980,
 1007, 1035, 1063, 1089, 1115,
 1137, 1147, 1168, 1177, 1198,
 1207, 1228, 1237, 1260, 1269,
 1293, 1302, 1323, 1332, 1353,
 1362, 1392, 1417, 1442, 1468, 1497
 \AskInc@MatchOption
 180, 184, 195, 242, 244, 247,
 332, 374, 377, 381, 387, 414, 415
 \AskInc@Next 372,
 393, 397, 604, 607, 610, 634, 638
 \AskInc@OrgInclude 110, 118
 \AskInc@OrgIncludeOnly
 161, 404, 412, 416, 421
 \AskInc@Pattern ... 505, 506, 509, 512
 \AskInc@PdfMatchAvailabletrue .. 177
 \AskInc@PdfMatchInit 445, 460
 \AskInc@PreviousAnswer
 236, 543, 551, 555, 562
 \AskInc@PreviousMatches
 262, 289, 290, 293
 \AskInc@PreviousMatchMethod
 240, 546, 557
 \AskInc@QuestionHeader . 254, 345, 579
 \AskInc@SanitizeAnswer . 208, 234, 370
 \AskInc@SaveAnswer 538, 587
 \AskInc@StringMakeMatch
 171, 180, 320, 376, 377, 442
 \AskInc@StringNoRegExp 170, 373
 \AskInc@StringPdfMatch
 172, 184, 244, 324, 379, 381
 \AskInc@Temp 278,
 279, 281, 452, 453, 455, 456,
 476, 477, 494, 496, 498, 506,
 562, 563, 565, 572, 573, 592, 601
 \AskInc@TimeoutFiles 261, 346
 \AskInc@TimeoutMainQuestion 336, 349
 \AskInc@TimeoutMatchMethod . 315, 348
 \AskInc@TimeoutMethod
 317, 320, 324, 330
 \AskInc@TimeoutPreviousAnswer ...
 302, 347
 \AskInc@WriteAddFile 103, 115
 \AskInc@WriteAddMatch 434, 532

| | |
|---|---|
| <code>\AskInc@WriteAux</code> | 58, 68, 92, 105, 123, 128, 154, 521, 534, 539 |
| <code>\AskInc@WriteDefAddFile</code> | 90, 104 |
| <code>\AskInc@WriteDefAddList</code> | 67, 91, 520 |
| <code>\AskInc@WriteDefAddMatch</code> | 519, 533 |
| <code>\AtBeginDocument</code> | 54, 57, 100, 121, 230, 529, 640, 711, 773, 1374 |
| <code>\AtBeginDocumentMode</code> | 659, 710, 1373 |
| <code>\AtEndDocument</code> | 754, 774 |
| <code>\AtEndOfPackage</code> | 120 |
| B | |
| <code>\begin</code> | 730, 755, 766, 785, 795, 801, 849, 852, 866, 874, 877, 891, 899, 902, 916, 924, 927, 941, 949, 952, 966, 974, 977, 993, 1001, 1004, 1021, 1029, 1032, 1049, 1057, 1060, 1075, 1083, 1086, 1101, 1109, 1112, 1127, 1135, 1144, 1158, 1166, 1174, 1188, 1196, 1204, 1218, 1226, 1234, 1250, 1258, 1266, 1283, 1291, 1299, 1313, 1321, 1329, 1343, 1351, 1359, 1378, 1386, 1389, 1403, 1411, 1414, 1428, 1436, 1439, 1453, 1462, 1465, 1482, 1491, 1494, 1511 |
| <code>\BOTTOM</code> | 749, 752 |
| C | |
| <code>\catcode</code> | 3, 4, 6, 7, 8, 12, 13, 14, 15, 16, 17, 18, 21, 22, 24, 25, 26, 27, 31, 33, 675, 683, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 814, 819 |
| <code>\chapter</code> | 644, 648, 652 |
| <code>\CheckAnswer</code> | 780, 848, 873, 898, 923, 948, 973, 1000, 1028, 1056, 1082, 1108, 1134, 1165, 1195, 1225, 1257, 1290, 1320, 1350, 1385, 1410, 1435, 1460, 1489 |
| <code>\CheckInclude</code> | 764, 867, 868, 869, 892, 893, 894, 917, 918, 919, 942, 943, 944, 967, 968, 969, 994, 995, 996, 1022, 1023, 1024, 1050, 1051, 1052, 1076, 1077, 1078, 1102, 1103, 1104, 1128, 1129, 1130, 1159, 1160, 1161, 1189, 1190, 1191, 1219, 1220, 1221, 1251, 1252, 1253, 1284, 1285, 1286, 1314, 1315, 1316, 1344, 1345, 1346, 1379, 1380, 1381, 1404, 1405, 1406, 1429, 1430, 1431, 1454, 1455, 1456, 1483, 1484, 1485, 1512, 1513, 1514 |
| <code>\CheckMatchMethod</code> | 790, 1461, 1490 |
| <code>\comma@list</code> | 431 |
| <code>\comma@normalize</code> | 430 |
| <code>\comma@parse@normalized</code> | 464, 501 |
| <code>\CompatibilityMode</code> | 662 |
| <code>\count@</code> | 671, 675, 677, 678, 681, 683, 684, 685 |
| <code>\csname</code> | 10, 175, 768 |
| D | |
| <code>\DeclareOption</code> | 146, 147, 148, 179, 183, 187, 194 |
| <code>\do</code> | 266, 280, 564, 572 |
| <code>\documentclass</code> | 665 |
| <code>\documentstyle</code> | 663 |
| E | |
| <code>\end</code> 732, 762, 770, 788, 798, 809, 851, 865, 876, 890, 901, 915, 926, 940, 951, 965, 976, 992, 1003, 1020, 1031, 1048, 1059, 1074, 1085, 1100, 1111, 1126, 1143, 1157, 1173, 1187, 1203, 1217, 1233, 1249, 1265, 1282, 1298, 1312, 1328, 1342, 1358, 1372, 1388, 1402, 1413, 1427, 1438, 1452, 1464, 1481, 1493, 1510, 1517 | |
| <code>\endcsname</code> | 10, 175, 768 |
| <code>\endinput</code> | 51 |
| <code>\endlinechar</code> | 5, 11, 23, 818 |
| <code>\EndTest</code> | 688, 721 |
| <code>\escapechar</code> | 210 |
| <code>\Expect</code> | 731, 756, 767, 786, 796, 807 |
| F | |
| <code>\filea</code> | 645 |
| <code>\fileb</code> | 649 |
| <code>\filec</code> | 653 |
| <code>\filec@ntents</code> | 812, 813 |
| G | |
| <code>\g@addto@macro</code> | 83, 292, 438, 511, 585, 782, 792, 811, 830 |
| <code>\gdef</code> | 81, 124, 129, 135, 137, 155, 436, 540, 583, 613, 618, 623, 781, 791, 828, 854, 855, 879, 880, 904, 905, 929, 930, 954, 955, 979, 980, 984, 989, 1006, 1007, 1011, 1014, 1017, 1034, 1035, 1039, 1042, 1045, 1062, 1063, 1067, 1088, 1089, 1093, 1114, 1115, 1119, 1136, 1137, 1146, 1147, 1167, 1168, 1176, 1177, 1197, 1198, 1206, 1207, 1227, 1228, 1236, 1237, 1241, 1246, 1259, 1260, 1268, 1269, 1273, 1276, 1279, 1292, 1293, 1301, 1302, 1322, 1323, 1331, 1332, 1352, 1353, 1361, 1362, 1391, 1392, 1416, 1417, 1441, 1442, 1467, 1468, 1496, 1497 |
| <code>\GetFileContents</code> | 802, 807, 808 |
| I | |
| <code>\if@filesw</code> | 60, 775 |
| <code>\if@partsw</code> | 141 |
| <code>\ifAskInc@Found</code> | 461, 466 |
| <code>\ifAskInc@PdfMatchAvailable</code> | 173, 182, 245, 323, 380 |
| <code>\ifcase</code> | 70, 478, 485, 824 |
| <code>\ifnum</code> | 677, 684 |
| <code>\ifx</code> | 70, 72, 74, 136, 152, 175, 244, 263, 267, |

| | |
|--|---|
| 271, 273, 281, 288, 289, 305, 332, 373, 376, 379, 390, 400, 407, 410, 415, 435, 442, 477, 506, 508, 551, 555, 557, 565, 570, 573, 581, 582, 602, 606, 609, 612, 617, 622, 662, 710, 729, 757, 768, 824, 825, 826, 1373 | \pdffilesize 804 |
| \immediate 61, 776 | \pdfmatch . 190, 325, 385, 478, 485, 496 |
| \include 111, 114, 265, 765 | \pdfunescapehex 803 |
| \includeonly 143, 162, 163, 656, 1374, 1376, 1524, 1532 | \ProcessOptions 149, 197 |
| \IncludeTests 724 | \providecommand 57, 69, 93, 109, 134, 522, 823, 835, 840 |
| J | \ProvidesPackage 52 |
| \jobname 807, 808, 849, 852, 874, 877, 899, 902, 924, 927, 949, 952, 974, 977, 1001, 1004, 1029, 1032, 1057, 1060, 1083, 1086, 1109, 1112, 1135, 1144, 1166, 1174, 1196, 1204, 1226, 1234, 1258, 1266, 1291, 1299, 1321, 1329, 1351, 1359, 1386, 1389, 1411, 1414, 1436, 1439, 1462, 1465, 1491, 1494 | R |
| | \RangeCatcodeInvalid 680, 689, 690, 691, 692 |
| | \renewcommand 114, 163 |
| | \repeat 679, 686 |
| | \RequirePackage 429, 450 |
| | \RestoreCatcodes .. 670, 673, 674, 704 |
| L | S |
| \LogTests 725 | \space 143, 165, 226, 318, 325, 332, 338, 368 |
| \loop 672, 682 | \Stack 751, 847, 872, 897, 922, 947, 972, 999, 1027, 1055, 1081, 1107, 1133, 1164, 1194, 1224, 1256, 1289, 1319, 1349, 1384, 1409, 1434, 1459, 1488 |
| M | T |
| \makeatletter 669, 726 | \Test 688, 707 |
| \makeatother 705 | \the 11, 12, 13, 14, 15, 16, 17, 18, 31, 675 |
| \MakeMatcher 453 | \TMP@EnsureCode 28, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50 |
| \MatchMethodText 791, 797 | \typein 367, 595, 727, 728 |
| \MessageBreak .. 143, 166, 189, 384, 628 | \typeout 255, 256, 257, 258, 259, 265, 287, 299, 303, 312, 313, 316, 328, 331, 337, 338, 340, 341, 342, 369, 599, 745, 746 |
| N | U |
| \NeedsTeXFormat 2 | \unexpanded 736, 741, 786, 787, 796, 797 |
| \newcommand 110, 161, 751, 764, 780, 790 | \usepackage 666, 667, 712, 716 |
| \newif 173, 461 | W |
| O | \write 61, 776 |
| \org@filecontents 812, 815 | X |
| \org@typein 727, 733 | \x 9, 21 |
| P | Z |
| \PackageWarning 164, 383 | \zap@space 225 |
| \PackageWarningNoLine 142, 188, 495, 627 | |
| \pdffiledump 804 | |