

# The phfsvnwatermark package<sup>1</sup>

Philippe Faist [philippe.faist@bluewin.ch](mailto:philippe.faist@bluewin.ch)

August 15, 2016

<sup>1</sup>This document corresponds to *phfsvnwatermark v1.0*, dated 2016/08/15. It is part of the *phfqitlx package suite*, see <https://github.com/phfaist/phfqitlx>.

---

phfsvnwatermark—Add a watermark on each page with version control information from SVN.

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>SVN Identification Methods</b>	<b>2</b>
2.1	SVN Keywords read with the <code>svn</code> package . . . . .	3
2.2	GIT information faked as SVN keywords and read with the <code>svn</code> package . . . . .	3
2.3	SVN Keywords read with the <code>svn-multi</code> package . . . . .	4
2.4	SVN Keywords read with the <code>svn-multi</code> package, with <code>curfile</code> support	4
<b>3</b>	<b>Placement of the Watermark</b>	<b>4</b>
3.1	Watermark on each page . . . . .	5
3.2	Manual placement . . . . .	5
<b>4</b>	<b>Implementation</b>	<b>5</b>
4.1	Common Formatting of ID tag . . . . .	6
4.2	Definitions for the different SVN info engines . . . . .	6
4.3	Placement Methods . . . . .	8
4.4	Package Options & Setup . . . . .	8
	<b>Change History</b>	<b>10</b>
	<b>Index</b>	<b>10</b>

---

## ■ 1 Introduction

This package allows you to add version control information as a gray watermark on each page of your document.

The watermark may be placed automatically on each page, or its placement may be controlled manually by the user.

It is based on reading SVN keywords from your source file (e.g. `$Id$`, which SVN expands upon the following commit to `$Id: file.tex ... $`). The tags are

read with either the `svn` or `svn-multi` packages (at your option).

There is also decent support for multiple files, with `svn-multi` and `currfile`. This allows to both display overall document version information as well as information for the current file.

## ■ 2 SVN Identification Methods

The SVN version of the input file(s) is read with the help of SVN keyword `$Id$`. (The `svn` package may also read `$Date$` and `$Author$`, I'm not sure any more [TODO: CHECK].)

Don't forget to enable source keyword expansion for your  $\LaTeX$  source(s). This is done with the help of the command:

```
svn propset svn:keywords "Id Date Author" my-file.tex
```

You must specify an identification method with the package option `id=(method)`.

`id=svn`

Read the keywords with the `svn` package. Keywords are set in the file with the command `\SVN $Id$`. (The `$Id$` keyword contains all the relevant information and only this one needs to be set.) See below.

`id=gitnotsvn`

Read the keywords with the `svn` package. However, a script is used to fake these tags from what are in fact GIT meta-information. This causes `$Id$` not to contain all the information, and the `$Date$` and `$Author$` tags must be read and included in the watermark, as well. See below.

`id=svn-multi`

Read the keywords with the `svn-multi` package. The ID information is set with the command `\svn{id}{$Id$}`. (The `$Id$` keyword contains all the relevant information.) See below.

`id=svn-multi-currfile`

Read the keywords with the `svn-multi` package, with `currfile` support. Ideal for a document with multiple source files which are `\input` into each other. The ID information is set in each file with the command `\svn{id}{$Id$}`. (The `$Id$` keyword contains all the relevant information.) See below.

In each case, the relevant SVN package is automatically loaded `svn`, `svn-multi` and/or `currfile`.

## 2.1 SVN Keywords read with the svn package

Probably the most straightforward solution. Just introduce in your  $\LaTeX$  source file, somewhere near the top, the line:

```
\SVN $Id$
```

Don't forget to enable source keyword expansion for this  $\LaTeX$  file, and next time you commit your file to SVN, the  $\$Id\$$  tag will be replaced by something like:

```
\SVN $Id: thesis.tex 1488 2016-05-03 16:19:34Z pfaist $
```

which identifies the file version.

## 2.2 GIT information faked as SVN keywords and read with the svn package

### WARNING

This method is buggy, and keywords won't always get updated. I'm currently seeking a better method. Unfortunately all methods for obtaining GIT meta-data require running GIT hooks or auxiliary scripts.

To force update the GIT keywords, make sure you file has no local modifications, delete the file and run `git checkout <file>`.

The idea is to use the same keyword parsing engine, with the `svn` package, but to plug in the GIT meta-information instead inside faked `svn` keyword tags.

In your source file, add the lines

```
\SVN $Id$  
\SVN $Date$  
\SVN $Author$
```

And then, with the additional GIT configuration described below, these keywords will be expanded to the hash of the file contents (and not the commit, this may be misleading!), the date of the commit and the author.

In order to set up GIT for this, add to the `.gitattributes` file (create it in the repository directory if necessary):

```
*.tex ident filter=rscs-keyword
```

Then copy the following code, and paste it at the end of your REPO/.git/config or HOME/.gitconfig:

```
[filter "rcs-keyword"]
clean = perl -pwe \"s/\\\\\\\\\\\\\\\\$(Date|Author)[^\\\\\\\\\\\\\\\\$]*\\\\\\\\\\\\\\\\$/\\\\\\\\\\\\\\\\$\\\\\\\\\\\\\\\\$1\\\\\\\\\\\\\\\\$/\"
smudge = "echo >&2 'Applying RCS filter rules on '%f'.' && perl -we \\
  \\\"\\$f=%f; use Date::Parse; use POSIX;\\
  chomp(\\$author=\\`git log -1 --format=format:%ae -- \\$f\\`);\\
  chomp(\\$gitdate=\\`git log -1 --format=format:%aD -- \\$f\\`);\\
  \\$date=POSIX::strftime('%Y-%m-%d %H:%M:%S %z (%a, %e %b %Y)',\\
  localtime(str2time(\\$gitdate)));\\
  print STDERR \\\"formatted date: \\$date\\\\\\\\n\\\\\\\\\";\\
  while (<>) {\\
    s/\\\\\\\\\\\\\\\\$Date[^\\\\\\\\\\\\\\\\$]*\\\\\\\\\\\\\\\\$/\\\\\\\\\\\\\\\\$Date: '\\$date.' \\$'/e;\\
    s/\\\\\\\\\\\\\\\\$Author[^\\\\\\\\\\\\\\\\$]*\\\\\\\\\\\\\\\\$/\\\\\\\\\\\\\\\\$Author: '\\$author.' \\$'/e;\\
    print \\$_; }\""
```

## 2.3 SVN Keywords read with the svn-multi package

The svn-multi package is an alternative, more powerful, package for parsing SVN keywords.

Add near the top of your files the command

```
\\svnid{\\$Id$}
```

and make sure keyword expansion is enabled in your source files.

## 2.4 SVN Keywords read with the svn-multi package, with currfil support

This method is the same as for the id=svn-multi method with the svn-multi package, but with added support for multiple files.

Now, the watermark displays *two* lines, with the overall document version information, and the version information of the current file.

# ■ 3 Placement of the Watermark

First, there is a master switch for enabling or disabling the watermark. In either case, the relevant SVN-related package is loaded anyway, to allow files to provide their meta-information regardless of whether the watermark is displayed or not.

```
watermark=true
```

Show the watermark with the ID information.

```
watermark=false
```

Do not show the watermark with the ID information. (Note that manually placed watermarks are not affected by this option.)

```
placement=shipout
```

Place the watermark at the bottom of each page;

```
placement>manual
```

Don't place the watermark anywhere automatically; the user is responsible for placing it with a call to `\phfsvnVersionIdTag` wherever they want.

### 3.1 Watermark on each page

With the `placement=shipout` option, the watermark is automatically added on each page.

```
\phfsvnShipoutWatermarkXposRight  
\phfsvnShipoutWatermarkYposBaseline
```

The macros `\phfsvnShipoutWatermarkXposRight` and `\phfsvnShipoutWatermarkYposBaseline` determine the position of the watermark. They are macros which expand to a length, not lengths themselves. You may redefine them to adjust the placement of the watermark.

### 3.2 Manual placement

You may also opt to place the watermark manually on the page with the `placement>manual` option. Do this if you want the information to appear for example in a header or footer.

```
\phfsvnVersionIdTag Places the ID tag immediately, at the given location. The starred version pro-  
\phfsvnVersionIdTag* duces a box of zero width.
```

## ■ 4 Implementation

Include useful packages.

```
1 \RequirePackage{kvoptions}  
2 \RequirePackage{calc}
```

make sure the 'color' or 'xcolor' package is loaded

```
3 \@ifpackageloaded{xcolor}{}{%
4   \@ifpackageloaded{color}{}{%
5     \RequirePackage{xcolor}%
6   }
7 }
```

## 4.1 Common Formatting of ID tag

common exterior formatting for Version ID tag.

```
8 \definecolor{phfsvnversionidcolor}{rgb}{0.6,0.6,0.6}
9 \def\phfsvnVersionIdTagOuterFont{\normalfont\scriptsize}
10 \def\phfsvnVersionIdTagInnerFont{\ttfamily}
11 \def\phfsvn@versionidtag{%
12   \begingroup%
13     \color{phfsvnversionidcolor}\phfsvnVersionIdTagOuterFont%
14     [\,\begingroup\phfsvnVersionIdTagInnerFont%
15       {\phfsvn@versionidtag@contents}\endgroup\,]%
16   \endgroup%
17 }
```

`\phfsvnVersionIdTag` For manual placement. Starred version does not smash the thing.

```
18 \newcommand\phfsvnVersionIdTag{%
19   \@ifstar\phfsvn@smashedsvnversionidtag\phfsvn@versionidtag
20 }
21 \def\phfsvn@smashedsvnversionidtag{%
22   \hspace*{0pt}\smash{\phfsvn@clap{\phfsvn@versionidtag}}
23 }
```

Helper macro<sup>1</sup>:

```
24 \def\phfsvn@clap#1{\hbox to 0pt{\hss#1\hss}}
```

## 4.2 Definitions for the different SVN info engines

`\phfsvn@versionidtag@contents@svn` Definitions for method id=svn:

`\phfsvn@doincludesvn@svn`

```
25 \def\phfsvn@versionidtag@contents@svn{%
26   \SVNId%
27 }
28 \def\phfsvn@doincludesvn@svn{
29   \RequirePackage{svn}
30 }
```

<sup>1</sup>see <https://www.tug.org/TUGboat/tb22-4/tb72perlS.pdf>

@versionidtag@contents@gitnotsvn  
@versionidtag@contents@gitnotsvn

Definitions for method id=gitnotsvn. Use the svn package because the files  
have fake SVN tags with in fact the GIT meta-info

```
31 \def\phfsvn@versionidtag@contents@gitnotsvn{%  
32   \SVNId\hspace*{1.5em}\SVNDate~\SVNTime\hspace*{1.5em}\SVNAuthor%  
33 }  
34 \def\phfsvn@doincludesvn@gitnotsvn{%  
35   \RequirePackage{svn}%  
36 }
```

n@versionidtag@contents@svnmulti  
\phfsvn@doincludesvn@svnmulti

Definitions for method id=svn-multi:

```
37 \def\phfsvn@versionidtag@contents@svnmulti{%  
38   SVN Document Version:\hspace*{1ex}%  
39   \svnmainfilename~r\svnrev~\svndate~\svnauthor%  
40 }  
41 \def\phfsvn@doincludesvn@svnmulti{%  
42   \PassOptionsToPackage{filehooks}{svn-multi}  
43   \RequirePackage{svn-multi}  
44 }
```

ersionidtag@for@svnmulticurrfile  
vn@doincludesvn@svnmulticurrfile

Definitions for method id=svn-multi-currfile:

```
45 \def\phfsvn@svnmulticurrfile@maxwidth{0.8\paperwidth}  
46 \newsavebox\phfsvn@box@upperline  
47 \newsavebox\phfsvn@box@lowerline  
48 \def\phfsvn@svnmulticurrfile@upperline{%  
49   SVN Document Version:\hspace{1.5ex}r\svnrev~\svndate~\svnauthor}%  
50 \def\phfsvn@svnmulticurrfile@lowerline{%  
51   \svnkw{Filename}:\hspace{1.5ex}r\svnfilerev~\svnfiledate~\svnfileauthor}%  
52 \def\phfsvn@versionidtag@for@svnmulticurrfile{%  
53   \begingroup%  
54   \color{phfsvnversionidcolor}%  
55   \phfsvnVersionIdTagOuterFont\phfsvnVersionIdTagInnerFont%  
56   \sbox\phfsvn@box@upperline{\phfsvn@svnmulticurrfile@upperline}%  
57   \sbox\phfsvn@box@lowerline{\phfsvn@svnmulticurrfile@lowerline}%  
58   \begin{minipage}[t]{\minof{\phfsvn@svnmulticurrfile@maxwidth}%  
59     {\maxof{\wd\phfsvn@box@upperline}{\wd\phfsvn@box@lowerline}}}%  
60   \parindent=0pt\relax\parskip=0pt\relax%  
61   \raggedleft%
```

Don't just \usebox the saved boxes, because if the filenames are long we want  
to re-layout and allow line breaks:

```
62   \par\phfsvn@svnmulticurrfile@upperline%  
63   \par\phfsvn@svnmulticurrfile@lowerline%  
64 \end{minipage}%  
65 \endgroup%  
66 }%  
67 \def\phfsvn@doincludesvn@svnmulticurrfile{%
```

```

68 \RequirePackage{currfile}%
69 \RequirePackage[filehooks]{svn-multi}%
70 }%

```

### 4.3 Placement Methods

`\phfsvn@doplace@shipout` The shipout placement method.

```

71 \def\phfsvn@doplace@shipout{%
72 \RequirePackage{eso-pic}
73 \AddToShipoutPicture{%
74 \setlength{\@tempdimb}{\phfsvnShipoutWatermarkXposRight}%
75 \setlength{\@tempdimc}{\phfsvnShipoutWatermarkYposBaseline}%
76 \setlength{\unitlength}{1pt}%
77 \put(2,\strip@pt\@tempdimc){%
78 \makebox(\strip@pt\@tempdimb,0)[r]%
79 {\hfill\phfsvn@versionidtag}}%
80 }%
81 }

```

`\phfsvnShipoutWatermarkXposRight` `\phfsvnShipoutWatermarkYposBaseline` Control the position of the watermark. These are proper macros which expand to a length; they are not declared as lengths directly. (This allows to calculate the lengths on the spot.)

```

82 \def\phfsvnShipoutWatermarkXposRight{0.9\paperwidth}
83 \def\phfsvnShipoutWatermarkYposBaseline{0.05\paperheight}

```

`\phfsvn@doplace>manual` There is nothing to do for the manual placement, since the info is placed manually anyway. This macro is called by the package option `treatment`.

```

84 \def\phfsvn@doplace>manual{}

```

### 4.4 Package Options & Setup

```

85 \SetupKeyvalOptions{
86 family=phfsvn,
87 prefix=phfsvn@
88 }

```

Setup code for a given ID method:

```

89 \def\phfsvn@SetupForId#1{%
90 \ifcsname phfsvn@SetupForId@#1\endcsname%
91 \csname phfsvn@SetupForId@#1\endcsname%
92 \else%
93 \PackageError{phfsvn}{Unknown SvnId method: '#1'}%
94 \fi
95 }

```



```

96 \def\phfsvn@doincludesvn{}
97 \def\phfsvn@SetupForId@svn{
98   \message{phfsvn: Using SvnId method = svn}
99   \let\phfsvn@doincludesvn\phfsvn@doincludesvn@svn
100  \let\phfsvn@versionidtag@contents\phfsvn@versionidtag@contents@svn
101 }
102 \def\phfsvn@SetupForId@gitnotsvn{
103   \message{phfsvn: Using SvnId method = gitnotsvn}
104   \let\phfsvn@doincludesvn\phfsvn@doincludesvn@gitnotsvn
105   \let\phfsvn@versionidtag@contents\phfsvn@versionidtag@contents@gitnotsvn
106 }
107 \expandafter\def\csname phfsvn@SetupForId@svn-multi\endcsname{
108   \message{phfsvn: Using SvnId method = svn-multi}
109   \let\phfsvn@doincludesvn\phfsvn@doincludesvn@svnmulti
110   \let\phfsvn@versionidtag@contents\phfsvn@versionidtag@contents@svnmulti
111 }
112 \expandafter\def\csname phfsvn@SetupForId@svn-multi-currfile\endcsname{
113   \message{phfsvn: Using SvnId method = svn-multi-currfile}
114   \let\phfsvn@doincludesvn\phfsvn@doincludesvn@svnmulticurrfile
115   % redefine whole versionidtag, not only contents:
116   \let\phfsvn@versionidtag\phfsvn@versionidtag@for@svnmulticurrfile
117 }
118 \def\phfsvn@SetupForId@{% no ID method
119   \PackageWarning{phfsvn}{*** No SvnId method provided, no watermark will be displayed.}
120   \phfsvn@watermarkfalse
121 }

```

Setup code for the placement:

```

122 \def\phfsvn@SetupPlacement#1{%
123   \ifcsname phfsvn@doplace@#1\endcsname%
124     \csname phfsvn@doplace@#1\endcsname%
125   \else%
126     \PackageError{phfsvn}{Unknown placement method: '#1'}%
127   \fi
128 }

```

Declare the keyval options:

```

129 \DeclareStringOption[] {id} {svn}
130 \DeclareBoolOption [true] {watermark}
131 \DeclareStringOption [shipout] {placement}
132 \DeclareDefaultOption{%
133   % We provide the standard LaTeX error.
134   \@unknownoptionerror
135 }

```

Process & execute options:

```

136 \ProcessKeyvalOptions*

```

Now, take action according to the given options.

Set up the correct ID method:

```
137 \phfsvn@SetupForId{\phfsvn@id}
```

Always include the relevant SVN package, so all files can have their meta-info tags regardless of whether the watermark is displayed or not:

```
138 \phfsvn@doincludesvn
```

Finish setting up, set up the watermark if we want it on.

```
139 \ifphfsvn@watermark
140 \phfsvn@SetupPlacement{\phfsvn@placement}
141 \fi
```

## Change History

v1.0

General: Initial version ..... 1

## Index

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

Symbols	A
<code>\,</code> ..... 14, 15	<code>\AddToShipoutPicture</code> ..... 73
<code>\@ifpackageloaded</code> ..... 3, 4	B
<code>\@ifstar</code> ..... 19	<code>\begin</code> ..... 58
<code>\@tempdimb</code> ..... 74, 78	<code>\begingroup</code> ..... 12, 14, 53
<code>\@tempdimc</code> ..... 75, 77	C
<code>\@unknownoptionerror</code> ..... 134	<code>\color</code> ..... 13, 54

\csname .....	91, 107, 112, 124	shipout .....	5
currfile .....	1, 2, 4	svn .....	2
<b>D</b>			
\DeclareBoolOption .....	130	svn-multi .....	2
\DeclareDefaultOption .....	132	svn-multi-currfile .....	2
\DeclareStringOption .....	129, 131	true .....	5
\definecolor .....	8	watermark .....	5
<b>E</b>			
\end .....	64	\PackageError .....	93, 126
\endcsname ..	90, 91, 107, 112, 123, 124	packages:	
\endgroup .....	15, 16, 65	currfile .....	1, 2, 4
\expandafter .....	107, 112	phfqitlx .....	1
<b>F</b>			
false (pkg. opt.) .....	5	phfsvnwatermark .....	1
<b>G</b>			
gitnotsvn (pkg. opt.) .....	2	svn .....	1-3, 7
<b>H</b>			
\hbox .....	24	svn-multi .....	1, 2, 4
\hfill .....	79	\PackageWarning .....	119
\hspace .....	22, 32, 38, 49, 51	\paperheight .....	83
\hss .....	24	\paperwidth .....	45, 82
<b>I</b>			
id (pkg. opt.) .....	2	\par .....	62, 63
\ifcsname .....	90, 123	\parindent .....	60
\ifphfsvn@watermark .....	139	\parskip .....	60
<b>L</b>			
\let .....	99, 100, 104, 105, 109, 110, 114, 116	\PassOptionsToPackage .....	42
<b>M</b>			
\makebox .....	78	phfqitlx .....	1
manual (pkg. opt.) .....	5	\phfsvn@box@lowerline ...	47, 57, 59
\maxof .....	59	\phfsvn@box@upperline ...	46, 56, 59
\message .....	98, 103, 108, 113	\phfsvn@clap .....	22, 24
\minof .....	58	\phfsvn@doincludesvn .....	
<b>N</b>			
\newsavebox .....	46, 47	.....	96, 99, 104, 109, 114, 138
\normalfont .....	9	\phfsvn@doincludesvn@gitnotsvn	
<b>P</b>			
package options:		.....	34, 104
false .....	5	\phfsvn@doincludesvn@svn ...	25, 99
gitnotsvn .....	2	\phfsvn@doincludesvn@svnmulti	
id .....	2	.....	41, 109
manual .....	5	\phfsvn@doincludesvn@svnmulti	
placement .....	5	.....	45, 114
		\phfsvn@doincludesvn@svnmulti .	37
		\phfsvn@doplace@manual .....	84
		\phfsvn@doplace@shipout .....	71
		\phfsvn@id .....	137
		\phfsvn@placement .....	140
		\phfsvn@SetupForId .....	89, 137
		\phfsvn@SetupForId@ .....	118
		\phfsvn@SetupForId@gitnotsvn .	102
		\phfsvn@SetupForId@svn .....	97
		\phfsvn@SetupPlacement ...	122, 140
		\phfsvn@smashedsvnversionidtag	
		.....	19, 21
		\phfsvn@svnmulti	
		.....	50, 57, 63
		\phfsvn@svnmulti	
		.....	45, 58

<code>\phfsvn@svnmulticurrfile@upperline</code>	<b>S</b>
..... 48, 56, 62	<code>\sbox</code> ..... 56, 57
<code>\phfsvn@versionidtag</code> .....	<code>\scriptsize</code> ..... 9
..... 11, 19, 22, 79, 116	<code>\setlength</code> ..... 74, 75, 76
<code>\phfsvn@versionidtag@contents</code>	<code>\SetupKeyvalOptions</code> ..... 85
..... 15, 100, 105, 110	<code>shipout (pkg. opt.)</code> ..... 5
<code>\phfsvn@versionidtag@contents@gitnotswash</code>	<code>\smash</code> ..... 22
..... 31, 31, 105	<code>\strip@pt</code> ..... 77, 78
<code>\phfsvn@versionidtag@contents@svn</code>	<code>svn</code> ..... 1–3, 7
..... 25, 100	<code>svn (pkg. opt.)</code> ..... 2
<code>\phfsvn@versionidtag@contents@svnmulti</code>	<code>svn-multi</code> ..... 1, 2, 4
..... 37, 110	<code>svn-multi (pkg. opt.)</code> ..... 2
<code>\phfsvn@versionidtag@for@svnmulticurrfile</code>	<code>svn-multi-currfile (pkg. opt.)</code> .... 2
..... 45, 116	<code>\SVNAuthor</code> ..... 32
<code>\phfsvn@watermarkfalse</code> .....	<code>\svnauthor</code> ..... 39, 49
..... 120	<code>\SVNDate</code> ..... 32
<code>\phfsvnShipoutWatermarkXposRight</code>	<code>\svndate</code> ..... 39, 49
..... 5, 74, 82	<code>\svnfileauthor</code> ..... 51
<code>\phfsvnShipoutWatermarkYposBaseline</code>	<code>\svnfiledate</code> ..... 51
..... 5, 75, 82	<code>\svnfilerev</code> ..... 51
<code>\phfsvnVersionIdTag</code> .....	<code>\SVNId</code> ..... 26, 32
..... 5, 18	<code>\svnkws</code> ..... 51
<code>\phfsvnVersionIdTag*</code> .....	<code>\svnmainfilename</code> ..... 39
..... 5	<code>\svnrev</code> ..... 39, 49
<code>\phfsvnVersionIdTagInnerFont</code> .	<code>\SVNTime</code> ..... 32
..... 10, 14, 55	
<code>\phfsvnVersionIdTagOuterFont</code> .	
..... 9, 13, 55	
<code>phfsvnwatermark</code> .....	<b>T</b>
..... 1	<code>true (pkg. opt.)</code> ..... 5
<code>placement (pkg. opt.)</code> .....	<code>\ttfamily</code> ..... 10
..... 5	
<code>\ProcessKeyvalOptions</code> .....	
..... 136	
<code>\put</code> .....	<b>U</b>
..... 77	<code>\unitlength</code> ..... 76
<b>R</b>	<b>W</b>
<code>\raggedleft</code> .....	<code>watermark (pkg. opt.)</code> ..... 5
..... 61	<code>\wd</code> .....
<code>\relax</code> .....	..... 59
..... 60	
<code>\RequirePackage</code> .....	
..... 1, 2, 5, 29, 35, 43, 68, 69, 72	