

The **svn-multi** package*

also known as **svnkw**

Martin Scharrer

martin@scharrer-online.de

<http://www.scharrer-online.de/latex/svn-multi>

December 3, 2008

1 Package naming

The authors first choice for this package was **svnkw** but the CTAN maintainer suggested a more descriptive name and put the package in the **svn-multi**¹ directory. In this version the style file is renamed to **svn-multi**, but a **svnkw** dummy style file is still provided for backward compatibility.

2 Introduction

This package lets you typeset keywords of the version control system Subversion² (svn), which is the successor of the popular CVS, inside your L^AT_EX files anywhere you like. Unlike the very similar package **svn**³ the usage of multiple files for one L^AT_EX document is well supported. The package acquires the keywords of the last changed file and provides them to the user through macros. The package has to read all keywords of all files first and writes the most recent values in an auxiliary file with an ‘.svn’ extension. This file is read back at the next L^AT_EX run which introduces a delay like by the table of contents. The standard L^AT_EX switch `\nofiles` can be used to suppress the file generation. Macros to typeset the keywords of the current **included** or **inputed** L^AT_EX file are also provided.

*This document corresponds to **svn-multi** v1.3b, dated 2008/12/03.

¹CTAN: <http://www.ctan.org/tex-archive/macros/latex/contrib/svn-multi/>

²Subversion homepage: <http://subversion.tigris.org/>

³CTAN: <http://www.ctan.org/tex-archive/macros/latex/contrib/svn/>

3 Usage

3.1 Including of the Subversion keywords

To include your Subversion Id keywords use `\svnId` or `\svnIdlong`. This macros should be written very early in each file, i.e. in the preamble of the main document soon after `\documentclass` and `\usepackage{svn-multi}` (or, outdated, `\usepackage{svnkW}`) and as first in *every* `\included` subfile before the `\chapter` macro. They do not create any output. See section 3.2 to learn how to typeset the keyword values.

`\svnId` Macro for the svn Id keyword. Write the macro as `\svnId{Id}` into your L^AT_EX files. A trailing colon with spaces after the `Id` is also valid but **everything else** except a valid Subversion string will cause a T_EX parse error. Don't forget to set the subversion property `svn:keywords` of the files to at least `'Id'`. Subversion will expand it at the next commit. Please note that because the value is read verbatim the macro should exactly be written like above. Spaces, newlines or comments between `\svnId` and the `{` will lead to T_EX parse errors.

`\svnIdlong` Macro for a "long Id". Saves similar values like in `'Id'` but from the keywords `HeadURL`, `LastChangedDate`, `LastChangedRevision` and `LastChangedBy`. The usage of `\svnId` or `\svnIdlong` is a matter of taste. The second is more readable inside the code and results in a nicer date and a full URL, not only the filename. Both can also be used together.

Write this macro like this (order of arguments not meaningful)

```
\svnIdlong
{$HeadURL$}
{$LastChangedDate$}
{$LastChangedRevision$}
{$LastChangedBy$}
```

in your files and set the subversion property `svn:keywords` of them to `'HeadURL LastChangedDate LastChangedRevision LastChangedBy'`.

Please note that the arguments are read verbatim. Special precaution are taken to allow spaces, newlines and comments direct after the `\svnIdlong` and after each of the four arguments, just in case someone need this. In fact everything not inside braces `{ }` is ignored.

The two macros above are sufficient for the operation of `svn-multi` but the following macros are provided for additional or alternative keyword handling.

`\svn` This macro let you typeset svn keywords directly. The only argument is the svn keyword in the usual dollars. The dollars will be stripped and the rest is typeset as normal text. The star version strips also the space before the last dollar. This macro alone was the very first version of `svnkW` and is still included for fast and simple keyword typesetting.

`\svnkWsave` This macro lets you include and save any keyword you like. The syntax is `\svnkWsave{${keyword}$}`. The keyword can be already expanded or not (no value and only `“:”` or nothing after the key name). This macro is also used internally and does not create any output. Please note that the argument is read

verbatim and that there should be no space between the macro and the argument's left brace.

3.2 Typesetting the keyword values

The following macros can be used to typeset the keyword values anywhere in the document. Please note that not all L^AT_EX fonts have all special characters, e.g. ‘_’ is not provided in the standard roman font. To properly typeset filename and URL containing these letters you can use either texttype font (`\texttt`) or use `{\urlstyle{rm}\svnnolinkurl{...}}` which needs the `hyperref` package.

Like already mentioned `svn-multi` knows two groups of keywords. The first group contains of the keywords for the whole document which hold the values of the most recent committed file, the second contains of the *current* or *file local* keywords, e.g. the keywords of the current file.

<code>\svnrev</code>	These macros hold the keyword values of the whole document, i.e. of the most recent revision. They can be used everywhere in every file of the L ^A T _E X document, after <code>\usepackage{svn}</code> of course. Please see section 3.2.1 how to typeset parts of the date.
<code>\svndate</code>	
<code>\svnauthor</code>	
<code>\svnfilerev</code>	These macros hold the keyword values of the current L ^A T _E X file, but only if it contains a <code>\svnid</code> or <code>\svnidlong</code> macro. Otherwise the macros hold still the values of the last file. Please see section 3.2.1 how to typeset parts of the date. See <code>\svnk</code> below for all other keywords.
<code>\svnfiledate</code>	
<code>\svnfileauthor</code>	
<code>\svnmainurl</code>	The macro <code>\svnmainurl</code> and <code>\svnmainfilename</code> hold the URL and the filename of the main L ^A T _E X file as long the keywords <code>HeadURL</code> or <code>Id</code> were used in it, respectively. These can be used to typeset this information anywhere in the document which might be more descriptive as the name of the current file (which can be typeset with <code>\svnk{HeadURL}</code> or <code>\svnk{Filename}</code> after <code>\svnid</code> or <code>\svnidlong</code> , respectively).
<code>\svnmainfilename</code>	
<code>\svnsetmainfile</code>	This will declare the current file as the main LaTeX file by defining the above macros. It will automatically be called at the end of the preamble so the user normally doesn't have to use it by him- or herself as long it isn't needed in the preamble.
<code>\svnk</code>	Please note that this macro changes the definition of <code>\svnmainurl</code> and <code>\svnmainfilename</code> directly without going over the auxiliary file. Calling it in several files will make this two macros inconsistent.
<code>\svnk</code>	All keywords saved with <code>\svnid</code> , <code>\svnidlong</code> or <code>\svnk</code> can be typeset by this macro which is a holdover of <code>pare-multi</code> times of this package. It takes one argument which must be a subversion keyword name. It then returns the current value of this keyword or nothing (<code>\relax</code>) when the keyword was not set yet. Examples:

```
\textsl{Revision: \svnk{Revision}}
```

```
URL: \url{\svnk{HeadURL}}
```

In the second example `\url` (`hyperref` package) is used to add a hyperlink and to avoid problems with underscores (`_`) inside the URL. `svn-multi` is also providing a macro `\svnnolinkurl` which works like `\url` but doesn't add a hyperlink. See the description of this macro for more details.

If the given keyword doesn't exist a package warning is given to allow spelling errors to be tracked down. This doesn't work well when `\svnkw` is used inside `\url`. In this case the warning code will be typeset(!) verbatim into the document by `\url`.

`\svnkwdef` This macro is used to define the keyword values, i.e. `\svnkwdef{<keyword>}{<value>}`. This is normally only called internally but could be used by the user to override single keywords. The values can then be typeset by `\svnkw`. Note that this macro has no influence on the calculation of the latest revision.

Note that for `\svnkw` and `\svnkwdef` all different names for one keyword are valid and result in the access of the same variable. So e.g. subversion treats `Rev`, `Revision` and `LastChangedRev` the same way and so does this macros. You can e.g. say `\svnkwdef{Rev}{123}` and then typeset it with `\svnkw{Revision}` or `\svnkw{LastChangedRev}` if you like.

3.2.1 Accessing Date Values

`\svnfileyear` Whenever the date information is read, i.e. by `\svnidlong`, `\svnid`, `\svnkwsave{Date}` or `\svnkwsave{LastChangedDate}`, the following macros are set to the appropriate date parts for the current file (the `\svnfile...` versions) and for the whole document.

`\svnfilemonth` Please note that the hour and timezone are dependent on the keyword which defines the date informations. The hour will be in UTC aka Zulu-time, i.e. timezone +0000, when the date comes from the `Id` keyword. Otherwise the hour and timezone will be in local time. To avoid confusion the `Id` and `Date/LastChangedDate` keywords, e.g. `\svnid` and `\svnidlong`, should not be intermixed and/or the timezone should always be typeset together with the time.

`\svnfileday` Please also note that the timezone macros only hold the sign and the first two digits of the timezone. This is needed to define `\svnpdfdate`. The last two digits are always '00' anyway. If you need full ISO timezone write `\svnfiletimezone00` or `\svntimezone00`.

`\svnfilehour`

`\svnfileminute`

`\svnfilesecond`

`\svnfiletimezone`

`\svnyear`

`\svnmonth`

`\svnday`

`\svnhour`

`\svnminute`

`\svnsecond`

`\svntimezone`

`\svnpdfdate` Returns the last changed date of the whole document in a format needed for `\pdfinfo`. Can be used like this: `\pdfinfo{ /CreationDate (D:\svnpdfdate) }`, to set the PDF creation date to the last changed date if you use `pdflatex` to compile your \LaTeX document.

3.3 Using full author names

If you like to have the full author⁴ names, not only the usernames, in your document you can use the following macros. First you have to register all authors of the document with `\svnRegisterAuthor` and then you can write e.g. `\svnFullAuthor{\svnauthor}` or `\svnFullAuthor{\svnfileauthor}`.

`\svnRegisterAuthor` The usage is `\svnRegisterAuthor{<username>}{<full name>}` which registers

⁴This means subversion authors, e.g. the persons who commit changes into the svn repository.

$\langle full\ name \rangle$ as full name for $\langle username \rangle$.

`\svnFullAuthor` Takes the username as argument and returns the full name if it was registered first with `\svnRegisterAuthor`, otherwise it returns the given username. The star version returns the username in parentheses after the full name.

3.4 Using full revision names

Like the author's also revision names/tags can be registered and used later. This macros were implemented on user request and have the drawback that you have to guess the next revision number of your document in order to get correct results when you like to tag the to-be-checked-in revision. Please note that this has nothing to do with the normal subversion tagging.

`\svnRegisterRevision` The usage is `\svnRegisterRevision{ $\langle revision\ number \rangle$ }{ $\langle tag\ name \rangle$ }` which registers $\langle tag\ name \rangle$ as tag name for $\langle revision\ number \rangle$.

`\svnFullRevision` Takes a $\langle revision\ number \rangle$ (i.e. `\svnrev`, `\svnfilerev` or a number) as argument and returns the full name if it was registered first with `\svnRegisterRevision`, otherwise it returns "Revision $\langle revision\ number \rangle$ ". The star version returns also the revision number leaded by 'r' in parentheses after the tag name, e.g. Name (r123).

3.5 Verbatim URLs with and without hyperlinks

`\svnnolinkurl` This macro allows you to write `\svnnolinkurl{\svnkw{HeadURL}}` and get the Head URL typeset verbatim. However `\url{\svnkw{HeadURL}}` (`hyperref` package) gives you the same result with hyperlinked. Both macros require the `hyperref` package which is not automatically loaded by `svn-multi`. Please load it manually when you like to use `\svnnolinkurl`.

Since v1.3 all keywords are read and typeset verbatim so this macro isn't this important anymore. However together with `hyperrefs` `\urlstyle` macro it can be used to have keyword values with special characters in roman font, which normally doesn't hold letters like '_'.

Please note that you can't use `hyperrefs` `\nolinkurl` because it won't expand `\svnkw`.

4 Implementation

`\svn` After *-testing, the intermediate macros `\svn@s` and `\svn@n` are called to strip the { } from `\svn[*]{ $\$. \dots$ }` and to remove the *. Then the actual macros are called to strip the dollars with or without the space before the last dollar.

```
1 \newcommand{\svn}{\@ifnextchar*{\svn@s}{\svn@n}}
2 \def\svn@n#1{\@svn@n#1}
3 \def\svn@s*#1{\@svn@s#1}
4 \def\@svn@n$#1$#1}
5 \def\@svn@s$#1 $#1}
```

4.1 Definition of init values

```

6 % Init values
7 \def\svnrev{0}      \def\svnfilerev{0}      \def\@svn@rev{0}
8 \def\svndate{}      \def\svnfiledate{}      \def\@svn@date{}
9 \def\svnauthor{}    \def\svnfileauthor{}    \def\@svn@author{}
10 \def\svnyear{0}     \def\svnfileyear{0}     \def\@svn@year{0}
11 \def\svnmonth{0}    \def\svnfilemonth{0}    \def\@svn@month{0}
12 \def\svnday{0}      \def\svnfileday{0}      \def\@svn@day{0}
13 \def\svnhour{0}     \def\svnfilehour{0}     \def\@svn@hour{0}
14 \def\svnminute{0}   \def\svnfileminute{0}   \def\@svn@minute{0}
15 \def\svnsecond{0}   \def\svnfilesecond{0}   \def\@svn@second{0}
16 \def\svntimezone{0} \def\svnfiletimezone{0} \def\@svn@timezone{0}
17 \def\svnmainurl{NOT SET}\def\svnmainfilename{NOT SET}

```

4.2 Id macros

4.2.1 Normal Id

`\svnId` Calls `\svnkwsave` with `\@svnidswtrue` so that the `Id` keyword will be parsed at the end of `\svnkwsave`.

```

18 \newcommand*{\svnId}{%
19 \@svnidswtrue
20 \svnkwsave
21 }
22 \newif\if@svnidsw
23 \@svnidswfalse

```

`\svn@scanId` Scans `svn Id` (after it got parsed by `\svnkwsave`). Awaits only `Id` value without leading ‘`Id:`’ and a trailing `\relax` as end marker. It calls `\@svn@scandate` to extract the date informations and `\@svn@updateid` to update global `Id` values and also sets the appropriate keywords.

```

24 \def\svn@scanId#1 #2 #3 #4 #5\relax{%
25 % #1 is filename, #2 is revision, #3 is date (JJJJ-MM-DD),
26 % #4 is time (HH:MM:SST), #5 is author (username)
27 \@svn@scandate{#3 #4}%
28 \@svn@updateid{#2}{#3 #4}{#5}%
29 \svnkwnode{Filename}{#1}%
30 \svnkwnode{Date}{#3 #4}%
31 \svnkwnode{Revision}{#2}%
32 \svnkwnode{Author}{#5}%
33 }

```

`\@svn@updateid` We first define the expanded arguments to variables for the user. The expansion is needed because the arguments content is mostly generic like `\svn@key` and `\svn@value` which can change very soon after this macro.

```

34 \def\@svn@updateid#1#2#3{% #1 = rev, #2 date, #3 author (username)
35 \xdef\svnfilerev{#1}%
36 \xdef\svnfiledate{#2}%
37 \xdef\svnfileauthor{#3}%

```

Then we check if the revision is non-empty (not yet expanded by subversion?) and larger then the current maximum value `\@svn@rev`. If yes we save all value to save them in the .svn-file later.

```

38 \ifx\svnfilerev\empty\else
39 \ifnum\@svn@rev<\svnfilerev
40 \xdef\@svn@rev{#1}%
41 \xdef\@svn@date{#2}%
42 \xdef\@svn@author{#3}%
43 \xdef\@svn@year{\svnfileyear}%
44 \xdef\@svn@month{\svnfilemonth}%
45 \xdef\@svn@day{\svnfileday}%
46 \xdef\@svn@hour{\svnfilehour}%
47 \xdef\@svn@minute{\svnfileminute}%
48 \xdef\@svn@second{\svnfilesecond}%
49 \xdef\@svn@timezone{\svnfiletimezone}%
50 \else\fi
51 \fi
52 }

```

4.2.2 Long Id

`\svnidlong` We clear the keyword value first to reduce the risk though bad user input.

```

53 \newcommand{\svnidlong}{%
54 \svnkwdef{HeadURL}{}%
55 \svnkwdef{LastChangedDate}{}%
56 \svnkwdef{LastChangedRevision}{0}%
57 \svnkwdef{LastChangedBy}{}%

```

The catcodes are changed by `\svn@catcodes` to allow \TeX -special characters inside the keywords. The braces `{ }` are changed to allow comments between the arguments. `\svnidlong@readargsfull` is called to read the arguments.

```

58 \begingroup
59 \svn@catcodes
60 \catcode'\{=12
61 \catcode'\}=12
62 \svnidlong@readargsfull
63 }

```

`\svn@catcodes` Changes all \TeX -special character to category “other”. The newline aka return is changed to category “ignore” so line breaks are not taken as part of the verbatim arguments.

```

64 \def\svn@catcodes{%
65 \let\do\@makeother \dospecials
66 \catcode'\^M9 \catcode'\ 10
67 \catcode'\{1 \catcode'\}2
68 }

```

`\svnidlong@readargsfull` Reads all four arguments of `\svnidlong` and passes them to `\svnidlong@readargs`. The normal argument braces are changed to category “other” and put into the

macros parameter text to remove all code between them. This is done to avoid problems with comments direct after one of the arguments. Because the braces are now non-special the parentheses are made to a local replacement.

```

69 \begingroup
70 \catcode'\{=12\catcode'\}=12
71 \catcode'\(=1\catcode'\)=2
72 \gdef\svnidlong@readargsfull#1{#2}#3{#4}#5{#6}#7{#8}{%
73 \svnidlong@readargs(#2)(#4)(#6)(#8)%
74 )
75 \endgroup

```

\svnidlong@readargs Calls sub macro for all four arguments and ends the catcode changes made by **\svnidlong**.

```

76 \def\svnidlong@readargs#1#2#3#4{%
77 \svnkwsave@read #1\relax
78 \svnkwsave@read #2\relax
79 \svnkwsave@read #3\relax
80 \svnkwsave@read #4\relax
81 \endgroup

```

Now the update macros for date and id are called.

```

82 \ifx\svnkWLastChangedDate\empty
83 \else\@svn@scanlongdate{\svnkWLastChangedDate}\fi
84 \@svn@updateid{\svnkW{LastChangedRevision}}{\svnkW{LastChangedDate}}
85 {\svnkW{LastChangedBy}}}%
86 \ignorespaces
87 }%

```

4.3 KeyWord Macros

\svnkwsave Save macro. Takes a dollar wrapped keyword string, reads it though **\svn@readkw** and saves it using **\svnkWdef**.

```

88 \def\svnkwsave{%
89 \begingroup
90 \svn@catcodes
91 \svnkwsave@readargs
92 }

```

\svnkwsave@readargs Reads full argument, calls parse submacro and ends catcode changes. If **\svnkwsave** was called by **\svnid** scans the id keyword by calling the scan macro.

```

93 \gdef\svnkwsave@readargs#1{%
94 \svnkwsave@read#1\relax
95 \endgroup
96 \if@svnidsw
97 \ifx\svnkWId\empty
98 \else
99 \expandafter
100 \svn@scanId\svnkWId\relax
101 \@svnidswfalse

```



```

102 \fi
103 \fi
104 \ignorespaces
105 }

\svnkwsave@read Reads the full keyword and strips the dollars.
106 \beginngroup
107 \catcode'\$=12
108 \gdef\svnkwsave@read $#1$\relax{%
109     \svn@checkcolon#1:\relax
110 }
111 \endgroup

\svnkwsave@parse Parse the keyword and save it away.
112 \beginngroup
113 \catcode'\$=11
114 \gdef\svnkwsave@parse$#1:#2${%
115 \expandafter\xdef\csname svnkw#1\endcsname{#2}}
116 \endgroup

\svnkwdef First we check if there is a ‘setter’-macro for the keyword called \svnkwdef@<keyword>.
117 \newcommand{\svnkwdef}[2]{%
118 \@ifundefined{svnkwdef@#1}
If not we call the general macro \svnkwdef@.
119 {\svnkwdef@{#1}{#2}}
If yes we just call it with the value as argument.
120 {\csname svnkwdef@#1\endcsname{#2}}%
121 }

\svnkwdef@ This macro defines the second argument under \svnkw<1st argument>. The \xdef
is used to expand the content first (needed for internal use) and make the definition
globally.
122 \newcommand{\svnkwdef@}[2]
123 {\expandafter\xdef\csname svnkw#1\endcsname{#2}}
Example: \svnkwdef{Revision}{23} will define \svnkwRevision as 23.

\svnkwdef@Rev ‘Setter’-macros for single keywords, used by \svnkwdef.
\svnkwdef@Author These are needed to have have a common value for all alternative keyword names
\svnkwdef@Date ala Rev, Revision, LastChangedRevision.
The keywords Rev, Author and Date are just calling \svnkwdef@ with a fixed
first argument.
124 \def\svnkwdef@Rev#1{\svnkwdef@{Rev}{#1}}
125 \def\svnkwdef@Author#1{\svnkwdef@{Author}{#1}}
126 \def\svnkwdef@Date#1{\svnkwdef@{Date}{#1}}

```

The long keywords are defined then as aliases of the short, first for writing

```
127 \let\svnkwdef@Revision=\svnkwdef@Rev
128 \let\svnkwdef@LastChangedRevision=\svnkwdef@Rev
129 \let\svnkwdef@LastChangedBy=\svnkwdef@Author
130 \let\svnkwdef@LastChangedAt=\svnkwdef@Date
```

and then for reading.

```
131 \def\svnkwRevision{\svnkwRev}
132 \def\svnkwLastChangedRevision{\svnkwRev}
133 \def\svnkwLastChangedBy{\svnkwAuthor}
134 \def\svnkwLastChangedAt{\svnkwDate}
```

So e.g. `\svnkw{LastChangedRevision}` is always be the same as `\svnkw{Rev}`.

We define default values for normal keywords. Keyword `Filename` is the name given by `Id` and not a real keyword.

```
135 \svnkwdef{Rev}{0}
136 \svnkwdef{Date}{}
137 \svnkwdef{Author}{}
138 \svnkwdef{Filename}{}
139 \svnkwdef{HeadURL}{}%
```

`\svnkw` Macro to get keyword value. Just calls `\svnkw{ARGUMENT}` where the argument interpreted as text. So e.g. `\svnkw{Date}` is the same as `\svnkwDate` but this could be changed later so always use this interface to get the keyword values.

```
140 \newcommand{\svnkw}[1]{%
141 \@ifundefined{svnkw#1}
142 {\PackageWarning{svn-multi}{SVN keyword '#1' not defined (typo?)}}
143 {\csname svnkw#1\endcsname}}%
```

4.4 Keyword check and strip macros

The following macros are used to test whether the given keywords are fully expanded or not. Subversion supports unexpanded keywords as input with or without colon and with or without trailing space(s), i.e. a: `KW`, b: `$KW:$` or c: `$KW: $`. To avoid \LaTeX syntax errors in this pre-commit state the keyword is checked by the following macros. Unexpanded keywords result in an empty value. Also leading and trailing spaces are removed.

`\svn@checkcolon` Checks if the keyword contains a colon. It is called by `\svnkwsave@read` with a trailing `:\relax` so that `#2` will be empty if there is no earlier colon or will hold the value with this trailing colon otherwise. The first case means that the keyword is unexpanded without colon (case a) which leads to an empty value. In the second case `\svn@stripcolon` is called to strip the colon and surrounding spaces. The final value is returned by `\svn@value`.

```
144 \def\svn@checkcolon#1:#2\relax{%
145 \def\svn@test{#2}%
```

```

146 \ifx\svn@test\empty
147 \svnkndef{#1}{}%
148 \else
149 \svn@stripcolon#2\relax
150 \svnkndef{#1}{\svn@value}%
151 \fi
152 }

```

\svn@stripcolon Strips the previous added colon (for **\svn@checkcolon**). The remaining argument is checked if it's empty (case b) or only a space (case c). Otherwise the keyword is expanded and **\svn@stripspace** is called to strip the spaces.

```

153 \def\svn@stripcolon#1:\relax{%
154 \def\svn@test{#1}%
155 \ifx\svn@test\empty
156 \gdef\svn@value{}%
157 \else
158 \ifx\svn@test\svn@spaceonly
159 \gdef\svn@value{}%
160 \else
161 \svn@stripspace#1\relax\relax
162 \fi\fi
163 }
164
165 \def\svn@spaceonly{ }

```

\svn@stripspace Strips leading space if present and calls **\svn@striptrailingspace** to strip the trailing space.

```

166 \def\svn@stripspace#1#2\relax{%
167 \def\svn@test{#1}%
168 \ifx\svn@test\svn@spaceonly
169 \gdef\svn@value{#2}%
170 \else
171 \svn@striptrailingspace#1#2\relax
172 \fi
173 }

```

\svn@striptrailingspace Strips trailing space using the macros parameter text. Must be called with **\relax** as end marker.

```

174 \def\svn@striptrailingspace#1 \relax{%
175 \gdef\svn@value{#1}%
176 }

```

4.5 Date Macros

\@svn@scandate Scans data informations in Id keyword and saves them in macros.

```

177 \def\@svn@scandate#1{\@svn@scandate@#1\relax}
178
179 \def\@svn@scandate@#1-#2-#3 #4:#5:#6#7#8\relax{%

```

```

180 \gdef\svnfileyear{#1}%
181 \gdef\svnfilemonth{#2}%
182 \gdef\svnfileday{#3}%
183 \gdef\svnfilehour{#4}%
184 \gdef\svnfileminute{#5}%
185 \gdef\svnfilesecond{#6#7}%
186 \gdef\svnfiletimezone{+00}% #8 always 'Z' for Zulu-time (UTC)
187 }

```

\svn@scanlongdate Scans data informations in Date keyword and saves them in macros.

```

188 \def\@svn@scanlongdate#1{\expandafter\@svn@scanlongdate@#1\relax}
189 %
190 \def\@svn@scanlongdate@#1-#2-#3 #4:#5:#6 #700 #8\relax{%
191 \gdef\svnfileyear{#1}%
192 \gdef\svnfilemonth{#2}%
193 \gdef\svnfileday{#3}%
194 \gdef\svnfilehour{#4}%
195 \gdef\svnfileminute{#5}%
196 \gdef\svnfilesecond{#6}%
197 \gdef\svnfiletimezone{#7}%
198 }

```

\svnpdfdate Returns date in a format needed for \pdfinfo.

```

199 \def\svnpdfdate{\svnyear\svnmonth\svnday
200 \svnhour\svnminute\svnsecond\svntimezone'00'}

```

4.6 Mainfile Makros

\svnsetmainfile Saves the current HeadURL and Filename keywords to macros. Will be called automatically in the preamble.

```

201 \newcommand{\svnsetmainfile}
202 {\edef\svnmainurl{\svnk{HeadURL}}
203 \edef\svnmainfilename{\svnk{Filename}}}
204 \AtBeginDocument{\svnsetmainfile}

```

4.7 Register and FullName Macros

\svnRegisterAuthor Saves the author's name by defining \svn@author@*(username)* to it.

```

205 \newcommand{\svnRegisterAuthor}[2]{%
206 \expandafter\def\csname svn@author@#1\endcsname{#2}%
207 }

```

\svnFullAuthor We test if the starred or the normal version is used and call the appropriate submacro \svnFullAuthor@star or \svnFullAuthor@normal.

```

\svnFullAuthor*
208 \newcommand{\svnFullAuthor}{%
209 \@ifnextchar{*}
210 {\svnFullAuthor@star}
211 {\svnFullAuthor@normal}
212 }

```

Both submacros are calling `\svnFullAuthor@` but with different arguments. The star macro also removes the star of course.

```
213 \def\svnFullAuthor@star*#1{\svnFullAuthor@{#1}{~(r#1)}}
214 \def\svnFullAuthor@normal#1{\svnFullAuthor@{#1}{}}
```

`\svnFullAuthor@` now sets the author's full name. Note that #2 is empty when the normal version is called.

```
215 \def\svnFullAuthor@#1#2{%
216 \ifundefined{svn@author@#1}
217 {#1}
218 {\csname svn@author@#1\endcsname #2}%
219 }
```

`\svnRegisterRevision` Saves the revision's name or tag by defining `\svn@revision@<revisionnumber>` to it.

```
220 \newcommand{\svnRegisterRevision}[2]{%
221 \expandafter\def\csname svn@revision@#1\endcsname{#2}
222 }
```

`\svnFullRevision` We test if the starred or the normal version is used and call the appropriate submacro `\svnFullRevision@star` or `\svnFullRevision@normal`.

`\svnFullRevision*`

```
223 \newcommand{\svnFullRevision}{%
224 \@ifnextchar{*}
225 {\svnFullRevision@star}
226 {\svnFullRevision@normal}
227 }
```

Both submacros are calling `\svnFullRevision@` but with different arguments. The star macro also removes the star of course.

```
228 \def\svnFullRevision@star*#1{\svnFullRevision@{#1}{~(r#1)}}
229 \def\svnFullRevision@normal#1{\svnFullRevision@{#1}{}}
```

`\svnFullRevision@` now sets the revision name. Note that #2 is empty when the normal version is called.

```
230 \def\svnFullRevision@#1#2{%
231 \ifundefined{svn@revision@#1}
232 {Revision #1}
233 {\csname svn@revision@#1\endcsname #2}%
234 }
```

`\svnnolinkurl` This code is taken from the `hyperref` package and is the definition of `\url` just without the part which creates the actual hyperlink. This needs of course the `hyperref` package. A warning is given if it isn't loaded.

```
235 \DeclareRobustCommand*\svnnolinkurl{%
236 \ifundefined{hyper@normalise}
237 {\PackageWarning{svn-multi}{Package hyperref is needed for \noexpand
238 \svnnolinkurl.}}
239 {\hyper@normalise\svnnolinkurl}}%
240 \def\svnnolinkurl@#1{\Hurl{#1}}
```

4.8 Auxiliary file generation and read-back

At the end of document we write the values to an auxiliary file.

```
241 \AtEndDocument{%
```

We first check if we have something to save. Revision, date and author must be non-empty. This suppresses the auxiliary file if the user doesn't use the appropriate macros but other provided by this package.

```
242 \if@filesw
243 \ifx\@svn@rev\empty\else
244 \ifnum\@svn@rev=0\else
245 \ifx\@svn@date\empty\else
246 \ifx\@svn@author\empty\else
```

Open outfile to write project keywords.

```
247 \newwrite\svn@write
248 \immediate\openout\svn@write=\jobname.svn
249 \immediate\write\svn@write{%
250 \@percentchar\space SVN cache^^J%
251 \noexpand\def\noexpand\svnrev{\@svn@rev}^^J%
252 \noexpand\def\noexpand\svndate{\@svn@date}^^J%
253 \noexpand\def\noexpand\svnauthor{\@svn@author}^^J%
254 \noexpand\def\noexpand\svnyear{\@svn@year}^^J%
255 \noexpand\def\noexpand\svnmonth{\@svn@month}^^J%
256 \noexpand\def\noexpand\svnday{\@svn@day}^^J%
257 \noexpand\def\noexpand\svnhour{\@svn@hour}^^J%
258 \noexpand\def\noexpand\svnminute{\@svn@minute}^^J%
259 \noexpand\def\noexpand\svnsecond{\@svn@second}^^J%
260 \noexpand\def\noexpand\svntimezone{\@svn@timezone}^^J%
261 }
262 \immediate\closeout\svn@write%
263 \fi\fi\fi\fi\fi%
264 }
265
```

Reread output from last compile run if it exists.

```
266 \InputIfFileExists{\jobname.svn}{}{}
```

Change History

v1.0		v1.2
General: Initial version	1	General: Renamed packet from <code>svnkw</code> to <code>svn-multi</code> to match CTAN directory. Wrapper file <code>svnkw.sty</code> is provided for backward compatibility.
v1.1		1
General: Added macros to extract and typeset date/time informations. Added macros to set and typeset main URL or filename.	1	\svn*: Added star version. Normal version was not changed to not

break existing documents with user defined keywords without leading space.	5	cial character like _ ^ \$ % & \. Rewrote keyword check macros to work with verbatim code. \nofiles is now obeyed.	1
\svnFullAuthor*: Macro now returns the username if the full name was not registered.	12	v1.3a	
\svnFullRevision*: New macro .	13	General: Fixed issue with unwanted spaces generated by \svnid, \svnidlong and \svnkwsave, e.g. when used in a file which is included with \input	1
\svnk*: Added warning when a wrong, maybe misspelled, keyword is given.	10		
\svnnolinkurl: New macro	13	v1.3b	
\svnRegisterRevision: New macro	13	General: Changed the way catcodes are modified to be compatible with the french option of the babel package or other packages which modify the list of special characters.	1
\svnsetmainfile: New macro ..	12		
v1.3			
General: Added verbatim support. Keywords can now contain spe-			

Index

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

Symbols		S
\\$ <i>107, 113</i>	\@svn@timezone <i>16, 49, 260</i>	\svn <i>1, 2</i>
\(..... <i>71</i>	\@svn@updateid <i>28, <u>34</u>, 84</i>	\svn* <i>1, 2</i>
\) <i>71</i>	\@svn@year . <i>10, 43, 254</i>	\svn@catcodes <i>59, <u>64</u>, 90</i>
\@svn@author <i>9, 42, 246, 253</i>	\@svnidswfalse . <i>23, 101</i>	\svn@checkcolon <i>109, <u>144</u></i>
\@svn@date <i>8, 41, 245, 252</i>	\@svnidswtrue <i>19</i>	\svn@n <i>1, 2</i>
\@svn@day .. <i>12, 45, 256</i>	\{ <i>60, 67, 70</i>	\svn@s <i>1, 3</i>
\@svn@hour . <i>13, 46, 257</i>	\} <i>61, 67, 70</i>	\svn@scanId <i><u>24</u>, 100</i>
\@svn@minute <i>14, 47, 258</i>	\^ <i>66</i>	\svn@spaceonly <i>158, 165, 168</i>
\@svn@month . <i>11, 44, 255</i>	_ <i>66</i>	\svn@stripcolon <i>149, <u>153</u></i>
\@svn@n <i>2, 4</i>		\svn@stripspace <i>161, <u>166</u></i>
\@svn@rev <i>7, 39, 40, 243, 244, 251</i>	D	\svn@striptrailingspace <i>171, <u>174</u></i>
\@svn@s <i>3, 5</i>	\do <i>65</i>	\svn@test <i>145, 146, 154, 155, 158, 167, 168</i>
\@svn@scandate . <i>27, <u>177</u></i>	\dospecials <i>65</i>	\svn@value ... <i>150, 156, 159, 169, 175</i>
\@svn@scandate@ <i>177, 179</i>	I	\svn@write <i>247, 248, 249, 262</i>
\@svn@scanlongdate <i>83, <u>188</u></i>	\if@svnidsw <i>22, 96</i>	
\@svn@scanlongdate@ <i>188, 190</i>	\ignorespaces .. <i>86, 104</i>	
\@svn@second <i>15, 48, 259</i>	N	\svnauthor ... <i>3, 9, 253</i>
	\newif <i>22</i>	\svndate <i>3, 8, 252</i>

\svnday . 4, 12, 199, 256	\svnFullRevision@star	\svnkWLastChangedDate
\svnfileauthor . 3, 9, 37 225, 228 82, 83
\svnfiledate ... 3, 8, 36	\svnhour 4, 13, 200, 257	\svnkWLastChangedRevision
\svnfileday	\svnid 2, 18 132
4, 12, 45, 182, 193	\svnidlong 2, 53	\svnkWRev 131, 132
\svnfilehour	\svnidlong@readargs	\svnkWRevision 131
4, 13, 46, 183, 194 73, 76	\svnkWsave ... 2, 20, 88
\svnfileminute	\svnidlong@readargsfull	\svnkWsave@parse .. 112
4, 14, 47, 184, 195 62, 69	\svnkWsave@read 77,
\svnfilemonth	\svnkW 3,	78, 79, 80, 94, 106
4, 11, 44, 181, 192	84, 85, 140, 202, 203	\svnkWsave@readargs
\svnfilerrev	\svnkWAuthor 133 91, 93
... 3, 7, 35, 38, 39	\svnkWDate 134	\svnmainfilename ..
\svnfilesecond	\svnkWdef ... 4, 29, 3, 17, 203
4, 15, 48, 185, 196	30, 31, 32, 54,	\svnmainurl .. 3, 17, 202
\svnfiletimezone ..	55, 56, 57, 117,	\svnminute 4, 14, 200, 258
4, 16, 49, 186, 197	135, 136, 137,	\svnmmonth 4, 11, 199, 255
\svnfileyear	138, 139, 147, 150	\svnnolinkurl ... 5, 235
4, 10, 43, 180, 191	\svnkWdef@ ... 119,	\svnnolinkurl@ 239, 240
\svnFullAuthor .. 5, 208	122, 124, 125, 126	\svnpdfdate 4, 199
\svnFullAuthor* . 5, 208	\svnkWdef@Author .. 124	\svnRegisterAuthor .
\svnFullAuthor@ ...	\svnkWdef@Date 124 4, 205
.... 213, 214, 215	\svnkWdef@LastChangedAt	\svnRegisterRevision
\svnFullAuthor@normal 130 5, 220
..... 211, 214	\svnkWdef@LastChangedBy	\svnrev 3, 7, 251
\svnFullAuthor@star 129	\svnsecond 4, 15, 200, 259
..... 210, 213	\svnkWdef@LastChangedRevision	\svnsetmainfile . 3, 201
\svnFullRevision 5, 223 128	\svntimezone
\svnFullRevision* 5, 223	\svnkWdef@Rev 124	... 4, 16, 200, 260
\svnFullRevision@ .	\svnkWdef@Revision . 127	\svnyear 4, 10, 199, 254
.... 228, 229, 230	\svnkWId 97, 100	
\svnFullRevision@normal	\svnkWLastChangedAt 134	
..... 226, 229	\svnkWLastChangedBy 133	