

The **svn-multi** package formerly known as **svnk**

Martin Scharrer

martin@scharrer-online.de

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

CTAN: <http://tug.ctan.org/pkg/svn-multi>

Version 2.0

March 23, 2009

1 Introduction

This package allows to typeset version control (VC) information provided by Subversion¹ keywords (e.g. `$Id: ... $`) in \LaTeX documents which can contain of multiple `.tex` files included using `\include` or `\input`. Subversion is a modern version control system designed to replace its predecessor CVS and uses integers as revision numbers.

This package reads the keywords of all files and provides the VC information of the most recent changed file of the document to the user through a set of macros. This information is written to an auxiliary `.svn` file during the first \LaTeX run and read back at the next which introduces the same delay known from the table of contents. The standard \LaTeX switch `\nofiles` can be used to suppress the file generation.

In addition to this basic functionality several more features are provided:

- Macros to typeset the VC information of the current source file.
- Access of all parts of the VC date.
- Formatting of author names or revision numbers.
- Definition of groups and subgroups.
- Including of the VC information of external files.
- Table of Revisions.

¹Subversion homepage: <http://subversion.tigris.org/>

1.1 Scope of Keywords

This package provides the Subversion keyword data in several different scopes: document-global, file-local and, new with v2.0, by group.

Document Global

The document global macros, like `\svnrev`, return the latest version control information (keyword data) for the whole multi-file document, i.e. the information of the latest changed file of the document. To collect, sort and provide this information is the main functionality of this package.

Local to Current File

There are also file-local macros, e.g. `\svnfilerev`, which return the version control information of the current file, i.e. the file they are used in. It is assumed here that every file using this macros calls first either a `\svnid` or `\svnidlong` macro or both. See section 2.2 for more details about the id macros. Please note that the file-local macros technically actually return the *last registered* information from the last `\svnid` or `\svnidlong`. As long the **filehooks** option (new in v2.0) is not enabled (explicit or implicit) this keyword macros will leak from one file over to the next. This will cause wrong results if they are used in a file before or without any id macros. With this option the macros will be reset at the beginning of every source file of the document.

Groups

Version 2.0 introduces the concept of groups. Several files of a multi-file L^AT_EX document can be grouped together and the latest version control information of all files of a group is provided by macros. This works in the same way as the global macros mentioned above but only with the files in the group. It can also be seen from the other side: the macros are local like the file-local macros mentioned above but for all files of the group, not only the current one.

These groups could also be called *file groups*, *keyword groups* or, like in programming languages, *namespaces*. In this manual they will be reference as simple *groups* most the time. In places where they could be confused with T_EX groups (`{ }`, `\begingroup \endgroup`), e.g. “in the current group” or “group local”, they will be called *keyword groups*.

There is no limitation (besides internal L^AT_EX resource limits) for the number of different groups. The files of one group do not have to be included in a row but can be included everywhere in the document. The version control information of the current group can be typeset with macros like `\svncgrev` (`cg` for *current group*). Also, a general but less robust macro `\svng{<group name>}{<key>}` is provided to access others groups by name everywhere in the document. To avoid some macro robustness problems the current group can be changed locally for the output macros using `\svnsetcg{<group name>}`.

See section 2.3 for further details and usage instructions on group macros.

2 Usage

The version control information are provided by Subversion keywords which first need to be read in by dedicated macros and can then be typeset using different macros.

2.1 Package Options

Since v2.0 this package provides options to enable only a needed features, e.g. to avoid problems with other packages or save T_EX memory. For backwards compatibility to pre-2.0 package versions all old features are enabled by default and all new features are disabled to save a little of T_EX memory.

All options except the first two are boolean key=value options (so far) and await either ‘true’ or ‘false’ as value. A missing value means ‘true’. So e.g. `[groups=true,verbatim=false,external]`, enables the **external** and **groups** options but disables the **verbatim** option.

The available options are:

old

Only pre-v2.0 features are active. This enables **verbatim** and disables all other options below. This is the default for reasons mentioned above.

all

Activates all features of the package.

verbatim

Controls the verbatim mode of the keyword parser macros. Normally verbatim mode is very much wanted to support strange characters in URLs and file names, but this options gives the user a possibility to disable verbatim, e.g. for trouble shooting. Please note that verbatim mode is needed in order to make **svn-multi** work with some packages, like **babel**² with the **french** option.

external

Controls the support for keywords from external files described in section 2.10. An old `.svn` file should be removed when this feature is deactivated to avoid *undefined macro* errors caused by macros placed there by this feature.

If the **groups** option is enabled the macro `\svnexternalgroup{<group name>}` can be used to declare a own group which is used for all the external files. Otherwise they are placed in the currently active group. This macro can be used several times during the document where an empty argument means *no group* and a ‘*’ means *current group*.

²CTAN: <http://tug.ctan.org/pkg/babel>

groups

Controls the keyword groups feature described in section 2.3.

subgroups

Controls the automatic declaration of all input files as subgroups so that there keyword information can be typeset inside other files. The group name is the file path without the file extension (`'subdir/filebase'`). It is possible to disable and re-enable this using `\svnsubgroupsfalse` and `\svnsubgroupstrue` during the document preamble or body to exclude certain files. See section 2.3.1 for additional information.

graphics

This option allows to automatically declare all images included using the macro `\includegraphics` from the `graphics`³/`graphicx`⁴ package as external files (see section 2.10). The options **external** and **autoload** are activated by this option so that the produced `.svx` files are loaded automatically. An `autoload=false` option after `graphics` will deactivate this, but then an `\svnexternal` macro must be included in all L^AT_EX files which should take the image revisions into account.

The `graphics` package is loaded if this option is active. If this package is needed with some special options it should be loaded by the L^AT_EX document before `svn-multi`.

Please note that this feature needs to tie itself into the `graphics` package and might fail if the internal structure of this package changes in future versions.

If the **groups** option is enabled the macro `\svngraphicsgroup{<group name>}` can be used to declare a own group which is used for all the graphic files (also for pgf images, see below). Otherwise they are placed in the group specified by `\svnexternalgroup` which defaults to the currently active group. This macro can be used several times during the document where an empty argument means *no group* and a `'*'` means *current group*.

Some graphics like logos can appear frequently in a document. Do not count them as part of each chapter they can be ignored using `\svnignoregraphic{<file path>}`. The macro `\svnconsidergraphic{<file path>}` disables this again. Such graphics can be then included manually using an explicit `\svnexternal` macro.

pgfimages

Identical to like the **graphics** option but for the `pgf`⁵ package (implemented against the version from 2008/01/15) with the `\pgfuseimage` and `\pgfimage` macros. Please also see the notes about package loading and ties mentioned above.

autoload

Controls automatic loading of corresponding `.svx` files at the begin of files included using `\input` or `\import`. This avoids the need of putting an `\svnexternal` macro

³CTAN: <http://tug.ctan.org/pkg/graphics>

⁴CTAN: <http://tug.ctan.org/pkg/graphicx>

⁵CTAN: <http://tug.ctan.org/pkg/pgf>

in every file just to load the `.svx` files created automatically by the **graphics** option. The option **external** is activated by **autoload**.

table

Controls the generation of a table of revisions which can be included using the `\tableofrevisions` macro. This table shows the revisions of all files and groups. This needs **groups** to work which is activated with **table**. Enable **subgroups** to include a list of all files per group. See the section 2.9 for more information.

filehooks

This option loads the `fink`⁶ package and installs at-begin-input-file and at-end-input-file hooks which are needed for many of the options above. While this option is enabled automatically if needed it can be also enabled manually to ensure that the file-local macros are reset to empty values at the begin of each input file. This prevents the keyword from leaking over from one file to the next. After every subfile the file-local keyword macros are also restored to the value of the parent file.

2.2 Including Subversion Keywords

Subversion keywords are included using `\svnid` or `\svnidlong`. These 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}` and as first in *every* subfile before an `\chapter` or similar macro. They do not create any output. See section 2.4 to learn how to typeset the keyword values.

`\svnid{Id$}`

The macro is for the `Id` keyword and must be written like shown. A trailing colon with or without spaces after the ‘`Id`’ is also valid but **everything else** except a valid Subversion string will cause a TeX parse error. The subversion property `svn:keywords` must be set on all source files and include ‘`Id`’ so that Subversion will expand it at the next commit.

`\svnidlong`

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

Macro for a “long Id”. Saves similar values like in ‘`Id`’ but from the above four keywords. 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. However, both can also be used together. In this case the `\svnid` macro should be come last. Because its revision is not higher (but identical) than

⁶CTAN: <http://tug.ctan.org/pkg/fink>

the revision of the `\svnidlone` macro it does not override its values. This way both the full time zone from the long and the file name from the short id macro can be accessed. Please note that all features from the 2.0 version load the `fink` package which lets you typeset the current file name anyway using `\finkbase.\finkext`⁷.

This macro must be written like seen above while the order of arguments is not meaningfull. The Subversion property `svn:keywords` must be set on all source files with an value which includes ‘`HeadURL LastChangedDate LastChangedRevision LastChangedBy`’ or one of their alternative spellings (e.g. ‘`URL`’, ‘`Rev`’, ‘`Date`’, etc.).

Please note that the arguments are read verbatim as long the `verbatim` option is not disabled explicitly. Special precaution are taken to allow spaces, newlines and comments direct after the `\svnidlone` and after each of the four arguments. In fact everything not inside braces `{ }` is ignored.

`\svn{<keyword>}`

`\svn*{<keyword>}`

This macro let you typeset svn keywords directly. 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 `svnk` and is still included for fast and simple keyword typesetting.

`\svnkwsave{<keyword>}`

This macro lets you include and save any keyword you like. 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.

2.3 Groups

Starting with v2.0 files can be grouped together and the keyword values of the latest revision of a group can be accessed. Use the `groups` option to activate these macros.

`\svngroup{<group name>}`

This macro declares all following files until the next `\svngroup` as part of the given keyword group. It can be placed inside the main file before some `\include/\input` macros or inside subfiles before the id macros, i.e. direct at the start of the file.

The changes done by this macro are \TeX global, i.e. there can’t be caught using \TeX groups (`{ }`). However, in order to prevent subfiles to change the group of the rest of the parent file the group will be restored to the previous one at the end of each input file.

⁷The file name in `Id` is always the original Subversion file name while the one given by the `fink` package is the current file name. Both could differ if the file got renamed.

The latest VC information of a group can be typeset with the `\svnvgXXX` macros or the `\svng` macro shown in section 2.4.

`\thesvngroup`

Returns the name of the current keyword group.

`\svnsetcg{<group name>}`

Normally the `\svncgXXX` macros mentioned below use the last keyword group defined by `\svngroup` but this can be changed using the `\svnsetcg` macro. The idea behind it is that the currently selected group can be changed locally to the current \TeX group for the keyword output macros `\svncgXXX` only while the group for the keyword input macros like `\svnid` is unaffected.

To reset the used group to the last one defined by `\svngroup` simply use `\svnsetcg` with an `*` as argument.

Example 1: `{\svnsetcg{abc}\svnFullAuthor{\svncgauthor}}`
would output the full author's name of group *abc*.

Example 2: To typeset the three keyword values of group *abc* somewhere outside this group use:

```
{\svnsetcg{abc}Rev: \svncgrev\\Date: \svncgdate\\  
Author: \svncgauthor\\}
```

Example 3: To typeset the date of group *abc* outside of this group in the format of `\today` use: `{\svnsetcg{abc}\svncgtoday}`

`\thesvncg`

Returns the name of the current group selected by `\svnsetcg`.

2.3.1 Files as Subgroups

The group feature could be used to access the version control information of single files anywhere in the document when these are defined as own groups for themselves. Because a file can only be in one group this would not be compatible with the normal usage of the group feature. Therefore a special feature was introduced to automatically or manually define a file as subgroup for itself which does not influence its membership in a normal group.

Declaration: This feature is enabled by the **subgroups** option. All files of the document are then automatically declared as extra groups. This can be disabled for parts of the document using `\svnsubgroupsfalse` and re-enabled using `\svnsubgroupstrue` macros. The current file can be manually declared as extra group with the `\svnsubgroup` macro.

`\svnsubgroup`

This macro declares the current file as subgroup. It is used automatically for every subfile if **subgroups** and `\svnsubgroupstrue` are enabled.

Exclude/Consider files extensions: The above mentioned automatically group declaration uses an hook which is triggered every time another file is read by the document. This unfortunately includes other packages, some auxiliary files and font, config and other files read in by this packages. An internal filter is in place to ignore this files by their file extension. This filter can be modified by the two following macros.

`\svnignoreextensions`{*<comma separated list of extension without leading dot>*}

Tells `svn-multi` to ignore the following file extension and never declare files with them as extra groups.

`\svnconsiderextensions`{*<comma separated list of extension without leading dot>*}

Tells `svn-multi` to (re-)consider the following file extension and declare files with them as extra groups if read in.

Typesetting: The keyword information of the subgroups (subfile including any included external files or subsubfiles) can be typeset using the normal group typeset macros mentioned below where the group name is the file path without extension. The keyword information of the `.tex` file alone can be typeset with the full file path including extension.

Example: `\svnsetcg{subdir/some_file.tex}\svncgrev` would typeset the revision of the file `some_file.tex` while `\svnsetcg{subdir/some_file}\svncgrev` would typeset the latest revision of the same file or any subfile or declared external file included by it.

2.4 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 proper typeset file names and URLs containing these letters you can use either teletype font (`\texttt`) or use `{\urlstyle{rm}\svnnolinkurl{...}}` which requires the **hyperref**⁸ package.

Like already mentioned `svn-multi` knows three scopes of keywords. The first contains of the keywords for the complete document which hold the values of the most recent committed file and the second contains of the *current* or *file local* keywords, e.g. the keywords of the current file. Only this two are described here while the third scope is described in section 2.3.

⁸CTAN: <http://tug.ctan.org/pkg/hyperref>

`\svnrev`

`\svndate`

`\svnauthor`

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^AT_EX document, after `\usepackage{svn}` of course. Please see section 2.5 how to typeset parts of the date.

`\svnfilerev`

`\svnfiledate`

`\svnfileauthor`

These macros hold the keyword values of the current L^AT_EX file, but only if it contains a `\svnid` or `\svnidlong` macro. Otherwise the macros hold either zero values or the values of the last file dependent on whether an option is enabled which enabled the `fink` package. Please see section 2.5 how to typeset parts of the date. See `\svnk` below for all other keywords.

`\svncgrev`

`\svncgauthor`

`\svncgdate`

These macros return keyword values of the currently selected keyword group. In order to hold them robust, which is important to use them in macros like `\svnFullAuthor`, they do not provide any arguments to select other groups than the current one. To access keyword values of other groups use the general macro `\svng` or change the locally selected keyword group using the macro `\svnsetcg`.

`\svng{<group name>}{<key>}`

This macro is a general form of the `\svncgXXX` macro mentioned above. The first argument is the requested keyword group, the second one the requested keyword in the form of `rev`, `date`, `author`, `year`, etc.. Please note that this macro can not be used inside macros like `\svnFullAuthor`.

`\svnmainurl`

`\svnmainfilename`

The macro `\svnmainurl` and `\svnmainfilename` hold the URL and the filename of the main L^AT_EX file as long the keywords `HeadURL` or `Id` 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 `\svnk{HeadURL}` or `\svnk{Filename}` after `\svnid` or `\svnidlong`, respectively).

`\svnsetmainfile`

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.

Please note that this macro changes the definition of `\svnmainurl` and `\svnmainfilename` directly without going over the auxiliary file. Calling it in several files will make this two macros inconsistent.

`\svnkw{<keyword name>}`

All keywords saved with `\svnid`, `\svnidlong` or `\svnkwsave` can be typeset by this macro which is a holdover from a very early version of this package when multiple files were not supported. It takes one argument which must be a subversion keyword name. It then returns the current value of this keyword or nothing (`\relax`) when the keyword was not set yet. Examples:

```
\textsl{Revision: \svnkw{Revision}}
URL: \url{\svnkw{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{<keyword name>}{<value>}`

This macro is used to define the keyword values. 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.

2.5 Accessing Date Values

<code>\svnyear</code>	<code>\svnfileyear</code>	<code>\svncgyear</code>
<code>\svnmonth</code>	<code>\svnfilemonth</code>	<code>\svncgmonth</code>
<code>\svnday</code>	<code>\svnfileday</code>	<code>\svncgday</code>
<code>\svnhour</code>	<code>\svnfilehour</code>	<code>\svncghour</code>
<code>\svnminute</code>	<code>\svnfileminute</code>	<code>\svncgminute</code>
<code>\svnsecond</code>	<code>\svnfilesecond</code>	<code>\svncgsecond</code>
<code>\svntimezone</code>	<code>\svnfiletimezone</code>	<code>\svncgtimezone</code>
<code>\svntimezonehour</code>	<code>\svnfiletimezonehour</code>	<code>\svncgtimezonehour</code>
<code>\svntimezoneminute</code>	<code>\svnfiletimezoneminute</code>	<code>\svncgtimezoneminute</code>

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

Please note that the hour and timezone are dependent on the keyword which defines the date information. 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. `\svn` and `\svnidl`, should not be intermixed and/or the timezone should always be typeset together with the time.

Starting with v1.4 of `svn-multi` the timezone macros return the full timezone, i.e. sign, hour and minute part, e.g. +0100, not only the sign and hour. The new macros `\svntimezonehour/\svnfiletimezonehour` and `\svntimezoneminute/\svnfiletimezoneminute` can be used to access only the hour including sign or the minute part, respectively.

Older versions of this manual assumed the minute part as always 00 and suggested to add it manually if needed: `\svnfiletimezone00` or `\svntimezone00`. In order not to “break” documents which followed this suggestion this two macros now remove a trailing 00 if present. However, this can be a problem when they are used inside an argument of another macro. One solution for this is to redefine them without the 00 removal part:

```
\renewcommand{\svntimezone}{\svntimezonehour\svntimezoneminute}
\renewcommand{\svnfiletimezone}{\svnfiletimezonehour\svnfiletimezoneminute}
```

To revert to the old (pre-v1.4) definition use:

```
\renewcommand{\svntimezone}{\svntimezonehour}
\renewcommand{\svnfiletimezone}{\svnfiletimezonehour}
```

`\svntime`

`\svnfiletime`

`\svncgtime`

This macros return the time part of the date only and simply return the corresponding hour, minute and second macros with a colon as separator.

`\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.

`\svntoday`

`\svnfiletoday`

`\svncgtoday`

These macros typeset the document-global, current-file or current-group date, respectively, using the format of `\today` which depends on the used language. To adjust the language of your document use the `babel` package.

2.6 Using Full Author Names

If you like to have the full author⁹ 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{<author>}{<full name>}`

This macro registers `<full name>` as full name for `<author>` (a subversion username) for later use with `\svnFullAuthor`.

`\svnFullAuthor{<author name or macro>}`

`\svnFullAuthor*{<author name or macro>}`

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. This is normally used in one of the following forms:

```
\svnFullAuthor{\svnauthor}  
\svnFullAuthor{\svnfileauthor}  
\svnFullAuthor{\svncgauthor}
```

2.7 Using Full Revision Names

Like the author's also revision names/tags can be registered and used later. These 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.

⁹This means subversion authors, e.g. the persons who commit changes into the svn repository.

`\svnRegisterRevision{<revision number>}{<tag name>}`

This registers *<tag name>* as tag name for *<revision number>* for later use with `\svnFullRevision`.

`\svnFullRevision{<revision number or macro>}`

`\svnFullRevision*{<revision number or macro>}`

Takes a revision number coming from a macro like `\svnrev`, `\svnfilerev` or a number as argument and returns the full name if it was registered first with `\svnRegisterRevision`, otherwise it returns “Revision *<revision number>*”. The star version returns also the revision number leaded by ‘r’ in parentheses after the tag name, e.g. Name (r123).

2.8 Verbatim URLs with and without Hyperlinks

`\svnnolinkurl{<macro with returns special text>}`

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 a hyperlink. 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 `hyperref`’s `\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 `hyperref`’s `\nolinkurl` because it won’t expand `\svnkw`.

2.9 Table of Revisions

Version 2.0 introduces this new feature which allows a overview table to be typeset which holds the version control information of all files of the document.

`\tableofrevisions`

If the **table** is enabled a table of revision is written into a file called *<main L^AT_EX file>.svt* (t for table) which can be included using the `\tableofrevisions` macro. The table contains the revision information of the complete document and of all defined groups. If the **subgroups** option is set the table will also include the single files sorted by group.

Table Format Macros

The `.svt` file is in an general format which uses a lot of macros to format the table and the different cells. This macros should not be used by the user directly but rather can be redefined to fit the personal taste. Figure 1 shows this macros in their position inside the table of revisions.

	\chapterORsection*{\svnrevisionsname}		
	\svnbeforetable		
	\svntable (like \begin{svntable})		
	\svntablehead		
\svnglobalrow	\svntabglobal		\svntabdate
\svngrouprow	\svntabgroup {\langle group \rangle}		{\langle year \rangle}{\langle month \rangle}{\langle day \rangle}
\svnsubgrouprow	\svntabsubgroup {\langle level \rangle}{\langle name \rangle}	\svntabrev{\langle rev \rangle}	{\langle hour \rangle}{\langle minute \rangle}{\langle second \rangle}
\svnfilerow	\svntabfile {\langle level \rangle}{\langle name \rangle}		{\langle TZ hour \rangle}{\langle TZ minute \rangle}
	\svntablefoot		
	\endsvntable (like \end{svntable})		
	\svnaftertable		

Figure 1: Table formatting macros and their position.

\svnrevisionsname

Holds the name of the table of revisions which is printed above it. The default is “Table of Revisions” which can be changed e.g. to another language.

\svnbeforetable

Can hold some content to be placed between the headline and the table.

\svnaftertable

Can hold some content to be placed after the table. By default it holds a macro to force a new page.

\svntable

\endsvntable

This macros build the table environment and hold a `\begin{tabular}{lllll}` `\end{tabular}` group by default. They can be redefined using

`\renewcommand{svntable}{...}{...}`.

In order to support special table packages like `tabularx`¹⁰ or `longtable`¹¹ the *content* of this macro is written verbatim into the `.svt` file. All other table macros are written as string and will be expanded when the `.svt` is read back.

\svntablehead

Holds the table head row “Name & Rev & Author & Date `\\hline`”.

¹⁰CTAN: <http://tug.ctan.org/pkg/tabularx>

¹¹CTAN: <http://tug.ctan.org/pkg/longtable>

`\svntablefoot`

Can hold the table foot row but is empty by default.

`\svnglobalrow` `\endsvnglobalrow`
`\svngrouprow` `\endsvngrouprow`
`\svnsubgrouprow` `\endsvnsubgrouprow`
`\svnfilerow` `\endsvnfilerow`

These macros are placed at the begin and end of every corresponding row and are empty by default. They can be used to insert horizontal lines before or after certain row types.

`\svntabglobal`

Simply typesets the name for the row holding the information of the whole document, e.g. “Document”.

`\svntabgroup`{*group name*}

Formats the group name.

`\svntabsubgroup`{*nesting level (1,2,3,...)*}{*subgroup name*}

Formats a subgroup name. The name is derived from a file name and therefore could contain characters like ‘_’ which should be taken into account. The macro `\svnnolinkurl` can be used for this. Because subgroups can be nested a number is provided to tell the nesting depth. This can be used to produce different indents for different levels: e.g.: `\addtolength{\leftskip}{#1\someindentamount}`.

`\svntabfile`{*nesting level (1,2,3,...)*}{*file path/name*}

Same as for the subgroup above but for file names.

`\svntabrev`{*revision number*}

Allows the formatting of the revision number. If empty the number will be typeset as normal. Conditional formatting is possible using the `ifthen`¹² package: e.g. the highest revision should be bold:

```
\renewcommand*{\svntabrev}[1]{\ifthenelse{#1=\svnrev}{\textbf{#1}}{#1}}
```

`\svntabauthor`{*author username*}

Formats the user names in the table. The macro `\svnFullAuthor` is a good candidate to be used here.

¹²CTAN: <http://tug.ctan.org/pkg/ifthen>

`\svntabdate{<year>}{<month>}{<day>}{<hour>}{<minute>}{<second>}{<TZ hour>}{<TZ minute>}`

Allows the formatting of the date. All values are given as numbers. The year has four digits and the timezone hour has a + or - sign, but all other values have only two digits.

2.10 Including Keywords of External Files

Version 2.0 now supports the inclusion of keywords of external files using the following macros and the Perl script `svn-multi.pl`.

`\svnexternal[<group name>]{{<filea>}{<fileb>}\dots{<filex>}}`

Subversion keywords of external files (e.g. non- \LaTeX files like images or even directories) can be included using this macro which awaits a list of files, each with the full path relative to the main \LaTeX file and each enclosed by `{ }`. The files must be under version control by Subversion, of course. Use `\svnexternalpath` to specify paths to be scanned for this files if they are not located relative to the main file. The requested filenames are written into the `.svn` auxiliary file and then processed by the external script `svn-multi.pl` which must be executed like described below. The appropriate keywords are then written in `<source file>.svx` files (x like eXternal) which are read in by the same `\svnexternal` macro at the next \LaTeX run. If a group name of `*` for the external files is specified using the optional argument the keywords will be placed in the same keyword group as this macro. The file local macros like `\svnfilerev` which appear in a source file after `\svnexternal` are affected, i.e. updated if one of the external revision is higher than the one of the source file. This makes sense if e.g. the included graphics are taken as logical part of a source file.

`\svnexternalpath{{<patha>}{<pathb>}\dots{<pathx>}}`

This macro can be used in the document preamble to declare a set of paths to be scanned for files specified with `\svnexternal`. This avoids the need to provide the path again and again for every file. The paths need to be enclosed in `{ }` and must be in Unix style, i.e. with `/` as directory separator and should end with a `/`. Windows users should just replace all `\` with `/`, e.g. `C:\My dir` gets `{C:/My dir/}`.

Script `svn-multi.pl`

The file `svn-multi.pl` which comes with the `svn-multi` package is an external Perl script which has to be run in the command line or by a \LaTeX development environment/editor like other tools like `BibTeX` or `Makeindex`. A Perl interpreter and a Subversion command line client (`svn`) must be installed to execute this script. Both are available for free for all major operating systems.

The script should be run inside the document folder in the following order:

1. Compile \LaTeX document, `.svn` file is generated.

2. Run `svn-multi.pl` script, `.svx` files are generated.
3. Compile \LaTeX document, `.svx` files are read in.

The script can be used with three different sets of arguments and with any combination of them. Please note that the word `jobname` stands for the main \LaTeX file name without the `.tex` extension.

`svn-multi.pl` $\langle\text{jobname}\rangle$

As already mentioned in the `\svnexternal` description above this script reads the requested external filename from the $\langle\text{jobname}\rangle$.`svn` file. The Subversion command line client `svn` is then used to fetch the needed keywords which are placed in a `\svnidlong` macro inside a $\langle\text{source file}\rangle$.`svx` file. Every single source file which uses `\svnexternal` will become its own `.svx` file which allows to attach specific external files to one (or more) specific source files.

`svn-multi.pl` $\langle\text{jobname}\rangle$ `--group` $\langle\text{group name}\rangle$ $\langle\text{file(s)}\rangle$...

The second way to use `svn-multi.pl` is to call it with a list of external files. A keyword group can be specified using the `--group` $\langle\text{group name}\rangle$ option which can be placed any number of times between the file names. The group is used for all external files listed after the option until the next group is specified. All keywords of these files are written in the $\langle\text{jobname}\rangle$.`svx` file and read in by the main \LaTeX file if a, possibly empty, `\svnexternal` macro is included. This allows for easy including of many external files without specifying them all inside the source file. For example `svn-multi.pl */*.jpg` (under Linux/Unix) will include the keywords of all JPG files in all subdirectories.

It is also possible to do this with a sub- \LaTeX -file by calling the script on it: `svn-multi.pl` $\langle\text{sub file}\rangle$ $\langle\text{external files for sub file}\rangle$, which will create/overwrite the $\langle\text{sub file}\rangle$.`svx` file. However the files given by `\svnexternal` in this sub-file will not be honoured in this case.

`svn-multi.pl` $\langle\text{jobname}\rangle$ `--group` $\langle\text{group name}\rangle$ `--fls`

Instead of providing a list of all non- \LaTeX /external files the `--fls` option can be used to read this list from the $\langle\text{jobname}\rangle$.`fls` file. This file is produced by the \LaTeX compiler when run with the `--recorder` option and contains a list of all input and output files. Only input files with a relative path are used. A corresponding keyword group can also be specified.

3 Compile Guide

A document which uses `svn-multi` needs to be compiled by \LaTeX in the following ways depending on the used features.

Basic Features – Global and Group Keywords

1. Compile document with \LaTeX . The `.svn` file is generated. Document global and group keyword macros are not valid yet.
2. Compile document again with \LaTeX . The `.svn` file is read by `svn-multi`. Document global and group keyword macros are now valid.

Table of Revisions

1. Compile document with \LaTeX . Document global and group keyword macros are calculated and written to the `.svt` file.
2. Compile document again with \LaTeX . The `.svt` file is read and typeset by the `\tableofrevisions` macro.

External Files

1. Compile document with \LaTeX . The `.svn` file is generated with references to the external files. Document global and group keyword macros are not valid yet. External files are not taken into account.
2. Run `svn-multi.pl` with the main base name (main file name without extension) as argument. This generates `.svx` files for each `.tex` file which used `\svnexternal`. These files contain the keywords of the external files.
3. Compile document again with \LaTeX . The `.svx` files are read by `svn-multi` and the `.svn` file is updated to take the new keywords into account. Document global and group keyword macros only hold internal values.
4. Compile document again with \LaTeX . The `.svn` file is read by `svn-multi`. Document global and group keyword macros are now fully valid.

4 Known Issues

This section lists some known issues of the `svn-multi` package and tries to provide some workaround. Please feel free to write `svn-multi` author if you detect any side effects or other issues caused by this package.

4.1 Packet listings uses `\input`

If a file `\langle basename \rangle.\langle extension \rangle` is typeset verbatim using `\lstinputlisting`, which uses `\input` to read the file, an existing `\langle basename \rangle.svx` file is also included as part of the listing. This can be avoided by code like this:

```
{\makeatletter\let\input\@input
\lstinputlisting[options]{filename}
}
```

5 Package Dependencies and Acknowledgements

This package uses some features from other packages and/or patches some macros of them to provide additional related features. This section is used to list this packages, their internal macro which got used and acknowledge the authors/maintainers of them. Please send error reports to the author of `svn-multi` and not to the people listed below.

All packages (including `svn-multi`) stand under the L^AT_EX Project Public Licence (LPPL) which can be found at <http://www.latex-project.org/lppl/> and can be freely downloaded from the Comprehensive TeX Archive Network (CTAN) at <http://www.ctan.org/>.

`hyperref`

The macro `\svnnolinkurl` is resembling the `hyperref` macro `\nolinkurl` and uses some its internal macros from the `\url` macro definition.

Used internal macros: `\hyper@normalise`, `\Hurl`

Version used: 2008/11/18 v6.78m

Licence: LPPL, any version

Authors/Maintainers: Sebastian Rahtz, Heiko Oberdiek

Location: CTAN: <http://tug.ctan.org/pkg/hyperref>

`fink`

The FiNK (File Name Keeper) package is used to get the file name and path of the input files. Two macros are patched to install hooks to execute own macros before and after each input file. This is only done if needed for an enabled option and therefore can be disabled using the option `old`.

Used internal macros: `\fink@file`, `\fink@nextdir`, `\fink@nexttext`, `\fink@nextbase`

Patched internal macros: `\fink@prepare`, `\fink@restore`

Version used: 2008/02/27 v2.1.1

Licence: LPPL, any version

Author/Maintainer: Didier Verna

Location: CTAN: <http://tug.ctan.org/pkg/fink>

`graphics`

If the `graphics` option is enabled the following macro is patched to record the file name and path of the included graphic.

Patched internal macros: `\Gin@setfile`

Version used: 2006/02/20 v1.0o

Licence: LPPL, any version

Author/Maintainer: David Carlisle, L^AT_EX3 Project
Location: CTAN: <http://tug.ctan.org/pkg/graphics>

pgf

Like the **graphics** package above a macro of this package is pathed to record the file names and paths of included images when the option **pgfimages** is enabled. Because this images pre-declares images for later use the internal declared ‘image macros’ are patched as well.

Patched internal macros: `\pgf@declareimage, \pgf@image@<image name>!`
Used internal macros: `\pgf@filename, \pgf@image`
Version used: 2008/01/15 v2.00
Licence: LPPL v1.3c and GPL v2
Author&Maintainer: Till Tantau
Location: CTAN: <http://tug.ctan.org/pkg/pgf>

latex

Parts of the macro definitions of the `\tableofcontents` macros from the **article** and **book** class of standard L^AT_EX were used to define a similar `\tableofrevisions` macro for both this classes and other similar classes.

Version used: 2005/09/16 v1.4f
Licence: LPPL v1.3c
Authors/Maintainers: L^AT_EX3 Project
Location: CTAN: <http://tug.ctan.org/pkg/latex>

6 Further Reading

The `svn-multi` package (in version 1.3) and its usage got discussed in the following articles:

- [1] Martin Scharrer, “Version Control of LaTeX Documents with `svn-multi`”, The PracT_EX Journal, (3), 2007. URL: <http://www.tug.org/pracjourn/2007-3/scharrer/>
- [2] Mark Eli Kalderon, “LaTeX and Subversion”, The PracT_EX Journal, (3), 2007. URL: <http://www.tug.org/pracjourn/2007-3/kalderon-svnmulti/>
- [3] Uwe Ziegenhagen, “LaTeX Document Management with Subversion”, The PracT_EX Journal, (3), 2007. URL: <http://www.tug.org/pracjourn/2007-3/ziegenhagen/>

7 Implementation

7.1 Package Header

Package Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
2 \ProvidesPackage{svn-multi}
3 [\filedate\space\fileversion\space SVN Keywords for multi-file LaTeX documents]
```

Options

Declaration of options and internal switches.

```
4 \RequirePackage{kvoptions}
5
6 \SetupKeyvalOptions{%
7   family = svn-multi,
8   prefix = @svnmulti@
9 }
10 \newif\if@svnmulti@anygraphic
11 \newif\if@svnmulti@autoload
12
13 \DeclareVoidOption{old}{%
14   \@svnmulti@verbatimtrue
15   \@svnmulti@groupsfalse
16   \@svnmulti@externalfalse
17   \@svnmulti@graphicsfalse
18   \@svnmulti@pgfimagesfalse
19   \@svnmulti@autoloadfalse
20   \@svnmulti@tablefalse
21   \@svnmulti@filehooksfalse
22   \@svnmulti@subgroupsfalse
23 }
24 \DeclareVoidOption{all}{%
25   \@svnmulti@verbatimtrue
26   \@svnmulti@groupstrue
27   \@svnmulti@externaltrue
28   \@svnmulti@graphicstrue
29   \@svnmulti@pgfimagestrue
30   \@svnmulti@autoloadtrue
31   \@svnmulti@tabletrue
32   \@svnmulti@filehookstrue
33   \@svnmulti@subgroupstrue
34 }
35 \DeclareBoolOption[true]{verbatim}
36 \DeclareBoolOption[false]{groups}
37 \DeclareBoolOption[false]{external}
38 \DeclareBoolOption[false]{subgroups}
39 \DeclareBoolOption[false]{graphics}
40 \DeclareBoolOption[false]{pgfimages}
41 \DeclareStringOption{autoload}[true]
```

```

42 \DeclareBoolOption[false]{table}
43 \DeclareBoolOption[false]{filehooks}
44
45 \ExecuteOptions{old}
46 \ProcessKeyvalOptions{svn-multi}
    Enable dependent options:
47 \def\svn@deoption#1{%
48   \csname if@svnmulti@#1\endcsname\else
49   \message{svn-multi: Required option '#1' enabled.}%
50   \csname @svnmulti@#1true\endcsname
51   \fi
52 }
53
54 \if@svnmulti@groups
55   \svn@deoption{filehooks}
56 \fi
57 \if@svnmulti@external
58   \svn@deoption{filehooks}
59 \fi
60 \if@svnmulti@subgroups
61   \svn@deoption{groups}
62   \svn@deoption{filehooks}
63 \fi
64 \if@svnmulti@graphics
65   \svn@deoption{external}
66   \svn@deoption{autoload}
67   \svn@deoption{filehooks}
68 \fi
69 \if@svnmulti@pgfimages
70   \svn@deoption{external}
71   \svn@deoption{autoload}
72   \svn@deoption{filehooks}
73 \fi
74 \if@svnmulti@autoload
75   \svn@deoption{external}
76   \svn@deoption{filehooks}
77 \fi
78 \if@svnmulti@table
79   \svn@deoption{groups}
80   \svn@deoption{filehooks}
81 \fi
    Check if autoload was set explicitly and obey the value.
82 \ifx\@svnmulti@autoload\@undefined
83 \else
84 \ifx\@svnmulti@autoload\empty
85 \else
86 \def\svn@temp{true}
87 \ifx\@svnmulti@autoload\svn@temp
88   \@svnmulti@autoloadtrue

```

```

89 \svn@depooption{external}
90 \svn@depooption{filehooks}
91 \else
92 \def\svn@temp{false}
93 \ifx\@svnmulti@autoload\svn@temp
94 \if@svnmulti@autoload
95 \PackageWarning{svn-multi}{Option 'autoload' disabled.}
96 \fi
97 \@svnmulti@autoloadfalse
98 \else
99 \PackageError{svn-multi}%
100 {Invalid value for 'autoload' option: '\@svnmulti@autoload'~J%
101 ! Only 'true','false' or empty (='true') are allowed!}
102 \fi\fi\fi\fi

General switch if any graphic option is enabled:

103 \if@svnmulti@graphics
104 \@svnmulti@anygraphictrue
105 \fi
106 \if@svnmulti@pgfimages
107 \@svnmulti@anygraphictrue
108 \fi

```

7.2 General Internal Macros

Some internal used macro which don't fit in any other section.

\svn@ifempty

#1: string

Tests if the given argument is empty. If so the first of the next two token will be expanded, the second one otherwise.

```

109 \def\svn@ifempty#1{%
110 \begingroup
111 \edef\svn@temp{#1}%
112 \ifx\svn@temp\empty
113 \endgroup
114 \expandafter
115 \@firstoftwo
116 \else
117 \endgroup
118 \expandafter
119 \@secondoftwo
120 \fi
121 }

```

\svn@ifequal

#1: string a

#2: string b

Tests if the given arguments are identical, e.g. same strings. If so the first of the next two token will be expanded, the second one otherwise.

```
122 \def\svn@ifequal#1#2{%
123   \begingroup
124   \edef\svn@stringa{#1}%
125   \edef\svn@stringb{#2}%
126   \ifx\svn@stringa\svn@stringb
127     \endgroup
128     \expandafter
129     \@firstoftwo
130   \else
131     \endgroup
132     \expandafter
133     \@secondoftwo
134   \fi
135 }
```

`\svn@ifvalidrev`

#1: macro name

Checks if the given macro (by name) is a valid revision, i.e. defined and greater than zero.

```
136 \def\svn@ifvalidrev#1{%
137   \begingroup
138   \@ifundefined{#1}%
139   {\def\svn@temp{-1}}%
140   {\expandafter\edef
141    \expandafter\svn@temp\expandafter{\csname #1\endcsname}}%
142   \ifnum\svn@temp>-1\relax
143     \endgroup
144     \expandafter
145     \@firstoftwo
146   \else
147     \endgroup
148     \expandafter
149     \@secondoftwo
150   \fi
151 }
```

`\svn@input`

#1: file name/path

Macro to load `.svx` and `.svt` files. The current keyword group is saved away and restored after the `.svx` file is loaded. The macros `\IfFileExists` with `\@@input` are used because `\InputIfFileExists` got redefined by the `fink` package and there is no need to use `fink` for this files.

```
152 \def\svn@input#1{%
153   \begingroup
```



```

154 \let\svn@rg\svn@g
155 \IfFileExists{#1}{\@@input #1\relax}{}%
156 \global\let\svn@g\svn@rg
157 \endgroup
158 }

```

\svn@inputsvx

#1: file name/path without extension

Used to save and restore file keywords when reading .svx files.

```

159 \def\svn@inputsvx#1{%
160   \svn@pushfilestack
161   \svn@input{#1.svx}%
162   \svn@popfilestack
163 }

```

\@namexdef

#1: macro name

Provides an expanding and global version of \@namedef.

```

164 \providecommand\@namexdef[1]{%
165   \expandafter\xdef\csname #1\endcsname
166 }

```

7.3 Definition of init values

Initialisation of at least the revision to a numeric value is necessary to not break the \ifnum tests later in this package. The revision is initialised to -1, but will be set to 0 if an unexpanded \$Rev:\$ keyword is read. This way it can be tested if a file had any keyword macros or not.

Note that there are two different macros for the document global keywords:

The user level \svn<kw> macros hold the global value and are only valid after a L^AT_EX run. They are initialised here and defined in the .svn file which is read at the end of the package if it exists and written at the end of the document.

The internal macros \@svn@<kw> store the oldest (i.e. highest revision) keywords read so far from the \svn_{id} and \svn_{id}long macros. They change during the document and are used to produce the values of the \svn<kw> macros when the .svn file is written.

Group wide macros are initialised when the group is first defined and have three different macros: \svng@<group>@<kw> (defined in .svn), \@svng@<group>@<kw> (accumulator) and also an access macro \svncg@<group> which uses \svn@g<current group>@<kw>.

```

167 % Init values
168 \def\svnrev{-1} \def\@svn@rev{-1}
169 \def\svndate{} \def\@svn@date{}
170 \def\svnauthor{} \def\@svn@author{}
171 \def\svnyear{0000} \def\@svn@year{0000}
172 \def\svnmonth{00} \def\@svn@month{00}

```

```

173 \def\svnday{00} \def\@svn@day{00}
174 \def\svnhour{00} \def\@svn@hour{00}
175 \def\svnminute{00} \def\@svn@minute{00}
176 \def\svnsecond{00} \def\@svn@second{00}
177 \def\svntimezonehour{+00} \def\@svn@timezonehour{+00}
178 \def\svntimezoneminute{00} \def\@svn@timezoneminute{00}
179 \def\svnmainurl{NOT SET} \def\svnmainfilename{NOT SET}
180 \def\svnurl{} \def\svnfname{}
181 \def\svn@temp{}
182
183 \def\svn@pg{} \def\svn@g{} \def\svn@cg{\svn@g}
184 \let\@svng@@files\relax
185
186 \def\svn@initfile{%
187 \gdef\svnfilerev{-1}%
188 \gdef\svnfiledate{}%
189 \gdef\svnfileauthor{}%
190 \gdef\svnfileyear{0000}%
191 \gdef\svnfilemonth{00}%
192 \gdef\svnfileday{00}%
193 \gdef\svnfilehour{00}%
194 \gdef\svnfileminute{00}%
195 \gdef\svnfilesecond{00}%
196 \gdef\svnfiletimezonehour{+00}%
197 \gdef\svnfiletimezoneminute{00}%
198 \gdef\svnfileurl{}%
199 \gdef\svnfilefname{}%
200 }
201 \svn@initfile

```

7.4 Timezone macros

\svntimezone

\svnfiletimezone

\svncgtimezone

These macros return the global, file-local and current group time zones, respectively. Since v1.4 the minute part is returned as well and the macro removes manually added 00 after it to support older documents.

```

202 \def\svntimezone{\svntimezonehour\svntimezoneminute\svn@gobblezeros}
203 \def\svnfiletimezone{\svnfiletimezonehour\svnfiletimezoneminute\svn@gobblezeros}
204 \def\svncgtimezone{\svncgtimezonehour\svncgtimezoneminute}

```

\svn@gobblezeros

\svn@gobblezeros@

This two cascaded macros remove a trailing 00 and are used by **\svnfiletimezone** and **\svntimezone**.

```

205 \def\svn@gobblezeros{%

```

```

206 \futurelet\svn@nextchar\svn@gobblezeros@
207 }
208 \def\svn@gobblezeros@{%
209 \let\@tempa=\relax
210 \def\@tempb{0}%
211 \ifx0\svn@nextchar
212 \let\@tempa=\@gobbletwo
213 \fi
214 \@tempa
215 }

```

\svntime

\svnfiletime

\svncgtime

This macros simple use the hour, minute and second macros.

```

216 \def\svntime{\svnhour:\svnminute:\svnsecond}
217 \def\svnfiletime{\svnfilehour:\svnfileminute:\svnfilesecond}
218 \def\svncgtime{\svncghour:\svncgminute:\svncgsecond}

```

7.5 *Today* macros

These macros use the `\today` macro to typeset the current date using the local language settings. Thanks and credit goes to Manuel Pégourié-Gonnard for suggesting this feature and for providing the code.

\svntoday

```

219 \newcommand*\svntoday{%
220 \beginngroup
221 \year\svnyear \month\svnmonth \day\svnday
222 \relax \today
223 \endgroup
224 }

```

\svnfiletoday

```

225 \newcommand*\svnfiletoday{%
226 \beginngroup
227 \year\svnfileyear \month\svnfilemonth \day\svnfileday
228 \relax \today
229 \endgroup
230 }

```

\svncgtoday

```

231 \newcommand*\svncgtoday{%
232 \@ifundefined{svng@\svn@cgs @year}{??}{%
233 \beginngroup
234 \year\svncgyear \month\svncgmonth \day\svncgday
235 \relax \today
236 \endgroup

```

```

237 }%
238 }%

```

7.6 Id macros

7.6.1 Normal Id

`\svnId`

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

```

239 \newcommand*{\svnId}{%
240   \@svnidswtrue
241   \svnkwsave
242 }
243 \newif\if@svnidsw
244 \@svnidswfalse

```

`\svn@scanId`

#1: file name
 #2: revision
 #3: date (YYYY-MM-DD)
 #4: time (HH:MM:SSZ)
 #5: author (username)

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 information and `\@svn@updateid` to update global Id values and also sets the appropriate keywords.

```

245 \def\svn@scanId#1 #2 #3 #4 #5\relax{%
246   \@svn@scandate{#3 #4}%
247   \@svn@updateid{#2}{#3 #4}{#5}{#1}%
248   \svnkwdef{Filename}{#1}%
249   \svnkwdef{Date}{#3 #4}%
250   \svnkwdef{Revision}{#2}%
251   \svnkwdef{Author}{#5}%
252 }

```

`\@svn@updateid`

#1: rev
 #2: date
 #3: author (username)
 #4: url

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

```

253 \def\@svn@updateid#1#2#3#4{%
254   \xdef\svnfilerev{#1}%

```

```

255 \xdef\svnfiledate{#2}%
256 \xdef\svnfileauthor{#3}%
257 \xdef\svnfileurl{#4}%
258 \svn@getfilename\svnfileurl%

```

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.

```

259 \ifx\svnfilerev\empty\else
260   \ifnum\@svn@rev<\svnfilerev
261     \xdef\@svn@rev{\svnfilerev}%
262     \xdef\@svn@date{\svnfiledate}%
263     \xdef\@svn@author{\svnfileauthor}%
264     \xdef\@svn@year{\svnfileyear}%
265     \xdef\@svn@month{\svnfilemonth}%
266     \xdef\@svn@day{\svnfileday}%
267     \xdef\@svn@hour{\svnfilehour}%
268     \xdef\@svn@minute{\svnfileminute}%
269     \xdef\@svn@second{\svnfilesecond}%
270     \xdef\@svn@timezonehour{\svnfiletimezonehour}%
271     \xdef\@svn@timezoneminute{\svnfiletimezoneminute}%
272     \xdef\@svn@url{\svnfileurl}%
273     \xdef\@svn@fname{\svnfilefname}%
274   \fi
275
276   \if@svnmulti@groups
277     \ifx\svn@g\empty\else
278       \svn@updategroup{\svn@g}%
279     \fi
280     \if@svnmulti@subgroups
281       \ifsvnsubgroups
282         \svn@updategroup{\svn@filedir\svn@filebase}%
283       \fi
284     \fi
285   \fi
286 \fi
287 }
288
289 \def\@svncg@save#1#2{%
290   \expandafter\xdef\csname @svng@\svn@g @#1\endcsname{#2}%
291 }
292

```

7.6.2 Long Id

\svnidlong

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

```

293 \newcommand{\svnidlong}{%
294   \svnkundef{HeadURL}{}%

```

```

295 \svnkwdef{LastChangedDate}{}%
296 \svnkwdef{LastChangedRevision}{0}%
297 \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.

```

298 \begingroup
299   \if@svnmulti@verbatim
300     \svn@catcodes
301     \catcode'\{=12
302     \catcode'\}=12
303   \else
304     \def\svnidlong@readargsfull{\svnidlong@readargs}%
305   \fi
306   \svnidlong@readargsfull
307 }

```

`\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.

```

308 \if@svnmulti@verbatim
309 \def\svn@catcodes{%
310   \let\do\@makeother
311   \dospecials
312   \catcode'\^~M9
313   \catcode'\ 10
314   \catcode'\{1
315   \catcode'\}2
316 }
317 \else
318   \def\svn@catcodes{}
319 \fi

```

`\svnidlong@readargsfull`

```

#1: some text, ignored
#2: Keyword 1
#3: some text, ignored
#4: Keyword 2
#5: some text, ignored
#6: Keyword 3
#7: some text, ignored
#8: Keyword 4

```

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.

```

320 \if@svnmulti@verbatim
321 \begingroup
322 \catcode'\{=12\catcode'\}=12
323 \catcode'\(=1\catcode'\)=2
324 \gdef\svnidlong@readargsfull#1{#2}#3{#4}#5{#6}#7{#8}{%
325 \svnidlong@readargs(#2)(#4)(#6)(#8)%
326 )
327 \endgroup
328 \fi

```

\svnidlong@readargs

#1: Keyword 1
 #2: Keyword 2
 #3: Keyword 3
 #4: Keyword 4

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

```

329 \def\svnidlong@readargs#1#2#3#4{%
330 \svnkwsave@read #1\relax
331 \svnkwsave@read #2\relax
332 \svnkwsave@read #3\relax
333 \svnkwsave@read #4\relax
334 \endgroup

```

Now the update macros for date and id are called.

```

335 \ifx\svnkwlLastChangedDate\empty\else
336 \@svn@scanlongdate{\svnkwlLastChangedDate}%
337 \fi
338 \@svn@updateid{\svnkwlLastChangedRevision}{\svnkwlLastChangedDate}}%
339 {\svnkwlLastChangedBy}{\svnkwlHeadURL}}%
340 \ignorespaces
341 }%

```

7.7 KeyWord Macros

\svnkwsave

Enabled verbatim mode and uses a sub macro to read the arguments afterwards.

```

342 \def\svnkwsave{%
343 \begingroup
344 \svn@catcodes
345 \svnkwsave@readargs
346 }

```

`\svnkwsave@readargs`

#1: \$kw: value\$

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.

```
347 \gdef\svnkwsave@readargs#1{%
348     \svnkwsave@read#1\relax
349 \endgroup
350 \if@svnidsw
351     \ifx\svnkwId\empty\else
352         \expandafter
353         \svn@scanId\svnkwId\relax
354     \@svnidswfalse
355 \fi
356 \fi
357 \ignorespaces
358 }
```

`\svnkwsave@read`

#1: keyword line without surrounding \$ \$

Reads the full keyword and strips the dollars.

```
359 \begingroup
360 \if@svnmulti@verbatim
361 \catcode'\$=12
362 \fi
363 \gdef\svnkwsave@read $#1$\relax{%
364     \svn@checkcolon#1:\relax
365 }
366 \endgroup
```

`\svnkwsave@parse`

#1: key

#2: value

Parse the keyword and save it away.

```
367 \begingroup
368 \catcode'\$=11
369 \gdef\svnkwsave@parse$#1:#2${%
370     \expandafter\xdef\csname svnkw#1\endcsname{#2}%
371 }%
372 \endgroup
```

`\svnkwnwdef`

#1: key

#2: value

First we check if there is a ‘setter’-macro for the keyword called `\svnkwnwdef@<keyword>`.

```
373 \newcommand{\svnkwnwdef}[2]{%
374     \@ifundefined{svnkwnwdef@#1}%
```


If not we call the general macro `\svnkwdef@`.

```
375 {\svnkwdef@{#1}{#2}}%
```

If yes we just call it with the value as argument.

```
376 {\csname svnkwdef@#1\endcsname{#2}}%  
377 }
```

`\svnkwdef@`

#1: key

#2: value

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.

```
378 \newcommand{\svnkwdef@}[2]{%  
379 \expandafter\xdef\csname svnkw#1\endcsname{#2}}%  
380 }
```

Example: `\svnkwdef{Revision}{23}` will define `\svnkwRevision` as 23.

`\svnkwdef@Rev`

`\svnkwdef@Author`

`\svnkwdef@Date`

`\svnkwdef@URL`

#1: value

‘Setter’-macros for single keywords, used by `\svnkwdef`.

These are needed to have have a common value for all alternative keyword names ala `Rev`, `Revision`, `LastChangedRevision`.

The keywords `Author` and `Date` are just calling `\svnkwdef@` with a fixed first argument. For the revision the value is checked if empty and then a 0 is substituted.

```
381 \def\svnkwdef@Rev#1{%  
382 \svn@ifempty{#1}%  
383 {\svnkwdef@{Rev}{0}}%  
384 {\svnkwdef@{Rev}{#1}}%  
385 }  
386 \def\svnkwdef@Author#1{\svnkwdef@{Author}{#1}}  
387 \def\svnkwdef@Date#1{\svnkwdef@{Date}{#1}}  
388 \def\svnkwdef@URL#1{\svnkwdef@{HeadURL}{#1}}
```

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

```
389 \let\svnkwdef@Revision=\svnkwdef@Rev  
390 \let\svnkwdef@LastChangedRevision=\svnkwdef@Rev  
391 \let\svnkwdef@LastChangedBy=\svnkwdef@Author  
392 \let\svnkwdef@LastChangedAt=\svnkwdef@Date
```

and then for reading.

```
393 \def\svnkwRevision{\svnkwRev}
```

```

394 \def\svnkwLastChangedRevision{\svnkwRev}
395 \def\svnkwLastChangedBy{\svnkwAuthor}
396 \def\svnkwLastChangedAt{\svnkwDate}
397 \def\svnkwURL{\svnkwHeadURL}

```

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.

```

398 \svnkwdef{Rev}{0}
399 \svnkwdef{Date}{}
400 \svnkwdef{Author}{}
401 \svnkwdef{Filename}{}
402 \svnkwdef{HeadURL}{}

```

`\svnkw`

#1: keyword name

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.

```

403 \newcommand{\svnkw}[1]{%
404   \@ifundefined{svnkw#1}%
405     {\PackageWarning{svn-multi}{SVN keyword '#1' not defined (typo?)}}%
406     {\csname svnkw#1\endcsname}%
407 }%

```

7.8 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 L^AT_EX 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`

#1: key

#2: potential value, might be empty

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`.

```

408 \def\svn@checkcolon#1:#2\relax{%
409   \svn@ifempty{#2}%
410     {\svnkwdef{#1}{}}%
411     {\svn@stripcolon#2\relax\svnkwdef{#1}{\svn@value}}%
412 }

```

\svn@stripcolon

#1: potential value

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@strip space** is called to strip the spaces.

```
413 \def\svn@stripcolon#1:\relax{%
414   \svn@ifempty{#1}%
415   {\gdef\svn@value{}}%
416   {\svn@ifequal{#1}{ }%
417    {\gdef\svn@value{}}%
418    {\svn@strip space#1\relax\relax}%
419   }%
420 }
```

\svn@strip space

#1: first character

#2: rest of string

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

```
421 \def\svn@strip space#1#2\relax{%
422   \svn@ifequal{#1}{ }%
423   {\gdef\svn@value{#2}}%
424   {\svn@strip trailing space#1#2\relax}%
425 }
```

\svn@strip trailing space

#1: string

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

```
426 \def\svn@strip trailing space#1 \relax{%
427   \gdef\svn@value{#1}%
428 }
```

\svn@gdef verb

#1: macro

```
429 \def\svn@gdef verb#1{%
430   \begingroup
431   \def\svn@temp{#1}%
432   \begingroup
433   \if@svnmulti@verbatim
434     \svn@catcodes
435     \fi
436   \svn@gdef verb@
437 }
```

\svn@defverb@

```
#1: verbatim stuff

438 \def\svn@gdefverb@#1{%
439     \endgroup
440     \expandafter\gdef\svn@temp{#1}%
441 \endgroup
442 }
```

\svn@namegdefverb

```
#1: macro name

443 \def\svn@namegdefverb#1{%
444     \begingroup
445     \expandafter\def
446     \expandafter\svn@temp
447     \expandafter{\csname #1\endcsname}%
448     \begingroup
449     \if@svnmulti@verbatim
450         \svn@catcodes
451     \fi
452     \svn@gdefverb@
453 }
```

7.9 Date Macros

\@svn@scandate

#1: date

Scans data information in Id keyword and saves them in macros.

```
454 \def\@svn@scandate#1{\@svn@scandate@#1\relax}
455
456 \def\@svn@scandate@#1-#2-#3 #4:#5:#6#7#8\relax{%
457     \gdef\svnfileyear{#1}%
458     \gdef\svnfilemonth{#2}%
459     \gdef\svnfileday{#3}%
460     \gdef\svnfilehour{#4}%
461     \gdef\svnfileminute{#5}%
462     \gdef\svnfilesecond{#6#7}%
463     \gdef\svnfiletimezonehour{+00}%
464     \gdef\svnfiletimezoneminute{00}% #8 always 'Z' for Zulu-time (UTC)
465 }
```

\@svn@scanlongdate

```
#1: Year
#2: Month
#3: Day
#4: Hour
#5: Minute
```

#6: Second
 #7: Timezone
 #8: Date description string (ignored)

Scans date information in Date keyword and saves them in macros.

```
466 \def\@svn@scanlongdate#1{\expandafter\@svn@scanlongdate@#1\relax}
467 %
468 \def\@svn@scanlongdate@#1-#2-#3 #4:#5:#6 #7 #8\relax{%
469   \gdef\svnfileyear{#1}%
470   \gdef\svnfilemonth{#2}%
471   \gdef\svnfileday{#3}%
472   \gdef\svnfilehour{#4}%
473   \gdef\svnfileminute{#5}%
474   \gdef\svnfilesecond{#6}%
475   \@svn@parsetimezone#7\relax%
476 }
```

\@svn@parsetimezone

#1: sign (+/-)
 #2: hour first digit
 #3: hour second digit
 #4: minute first digit
 #5: minute second digit

Scans timezone and splits hour and minute part.

```
477 \def\@svn@parsetimezone#1#2#3#4#5\relax{%
478   \gdef\svnfiletimezonehour{#1#2#3}%
479   \gdef\svnfiletimezoneminute{#4#5}%
480 }
```

\svnpdfdate

Returns date in a format needed for \pdfinfo.

```
481 \def\svnpdfdate{%
482   \svnyear\svnmonth\svnday
483   \svnhour\svnminute\svnsecond\svntimezonehour'\svntimezoneminute'%
484 }
```

7.10 Mainfile Makros

\svnsetmainfile

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

```
485 \newcommand{\svnsetmainfile}{%
486   \xdef\svnmainurl{\svnfileurl}%
487   \xdef\svnmainfilename{\svnfilename}%
488 }
489 \AtBeginDocument{\svnsetmainfile}
```

7.11 Register and FullName Macros

`\svnRegisterAuthor`

#1: author username

#2: Full Name

Saves the author's name by defining `svn@author@<username>` to it.

```
490 \newcommand{\svnRegisterAuthor}[2]{%
491   \expandafter\def\csname svn@author@#1\endcsname{#2}%
492 }
```

`\svnFullAuthor`

`\svnFullAuthor*`

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

```
493 \newcommand{\svnFullAuthor}{%
494   \@ifnextchar*{%
495     {\svnFullAuthor@star}%
496     {\svnFullAuthor@normal}%
497   }%
```

`\svnFullAuthor@star`

#1: username

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

```
498 \def\svnFullAuthor@star*#1{%
499   \edef\svn@temp{#1}%
500   \svnFullAuthor@{\svn@temp}{~(\svn@temp)}%
501 }%
```

`\svnFullAuthor@normal`

#1: username

```
502 \def\svnFullAuthor@normal#1{%
503   \edef\svn@temp{#1}%
504   \svnFullAuthor@{\svn@temp}{}%
505 }%
```

`\svnFullAuthor@`

#1: username

#2: previous defined trailing string

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

```
506 \def\svnFullAuthor@#1#2{%
507   \@ifundefined{svn@author@#1}%
508     {#1}%
509     {\csname svn@author@#1\endcsname #2}%
510 }
```

`\svnRegisterRevision`

#1: revision number

#2: tag name

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

```
511 \newcommand{\svnRegisterRevision}[2]{%
512   \expandafter\def\csname svn@revision@#1\endcsname{#2}%
513 }
```

`\svnFullRevision`

`\svnFullRevision*`

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

```
514 \newcommand{\svnFullRevision}{%
515   \@ifnextchar{*}%
516     {\svnFullRevision@star}%
517     {\svnFullRevision@normal}%
518 }
```

`\svnFullRevision@star`

#1: revision number

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

```
519 \def\svnFullRevision@star*#1{%
520   \edef\svn@temp{#1}%
521   \svnFullRevision@\svn@temp{~(r\svn@temp)}%
522 }
```

`\svnFullRevision@normal`

#1: revision number

```
523 \def\svnFullRevision@normal#1{%
524   \edef\svn@temp{#1}%
525   \svnFullRevision@\svn@temp{}%
526 }
```

`\svnFullRevision@`

#1: revision number

#2: previous defined trailing string

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

```
527 \def\svnFullRevision@#1#2{%
528   \@ifundefined{svn@revision@#1}%
529     {Revision #1}%
530     {\csname svn@revision@#1\endcsname #2}%
531 }
```

7.12 Input File Name

The **fink** package is used to get the input file names. AtBegin/AtEnd hooks are installed which will be used later.

```
532 \if@svnmulti@filehooks
```

Load **fink** package and check if all needed macros are provided.

```
533 \RequirePackage{fink}[2008/02/27]
534 \begingroup
535 \def\svn@finkerror{%
536 \PackageError{svn-multi}{Your installed version of the 'fink' package does not
537 provide the needed macros. It is either too old or too new.
538 Try a different version, e.g. v2.1.1 from 2008/02/27}{}}%
539 \let\svn@finkerror\relax
540 }
541 \@ifundefined{finkpath}{\svn@finkerror}{}%
542 \@ifundefined{finkdir}{\svn@finkerror}{}%
543 \@ifundefined{finkbase}{\svn@finkerror}{}%
544 \@ifundefined{fink@prepare}{\svn@finkerror}{}%
545 \@ifundefined{fink@restore}{\svn@finkerror}{}%
546 \@ifundefined{fink@maindir}{\svn@finkerror}{}%
547 \@ifundefined{fink@mainext}{\svn@finkerror}{}%
548 \endgroup
```

\svn@removedotslash

#1: string (e.g. file path) which might start with ./

Removes leading './' from given macro (holding a directory path). Awaits a macro as argument which is redefined inside the current group!

```
549 \def\svn@removedotslash#1{%
550 \def\svn@removedotslash##1##2##3\relax{%
551 \svn@ifequal{./}{##1##2}%
552 {\edef#1{##3}\def\next{\svn@removedotslash@##3\empty\empty\empty\relax}}%
553 {\edef#1{##1##2##3}\let\next\relax}%
554 \next
555 }%
556 \expandafter\svn@removedotslash@#1\empty\empty\empty\relax
557 }
```

Init values for file name macros.

```
558 \let\svn@mainext\fnk@mainext
559 \let\svn@maindir\fnk@maindir
560 \svn@removedotslash\svn@maindir
561 \edef\svn@filebase{\jobname}%
562 \edef\svn@fileext{\svn@mainext}%
563 \edef\svn@filedir{\svn@maindir}%
```

Filename and -path are build using the other macros:

```
564 \def\svn@filename{\fink@file\svn@filebase\svn@fileext}%
565 \def\svn@filepath{\svn@filedir\svn@filename}%
```


`\svnmulti@begininputfilehook`

This hook is installed in the `\fink@prepare` macro from the `fink` package which will be executed at the begin of a input file. The file name and path are not yet in `\finkpath` etc. but in `\fink@nextpath`.

```
566 \def\svnmulti@begininputfilehook{}
567 \message{Package svn-multi: patching macro '\string\fink@prepare' from the
568 'fink' package!}%
569 \let\svnmulti@fink@prepare\fink@prepare
570 \renewcommand*{\fink@prepare}[1]{%
571   \svnmulti@fink@prepare{#1}%
572   \if@svnmulti@groups
573     \svn@ifequal{\svn@filepath}{\jobname.\svn@mainext}%
574     {\global\let\svn@pg\svn@g}%
575     {\xdef\svn@pg{\svn@filedir\svn@filebase}}%
576   \fi
577   \xdef\svn@filebase{\fink@nextbase}%
578   \xdef\svn@fileext{\fink@nextext}%
579   \xdef\svn@filedir{\fink@nextdir}%
580   \svn@removedotslash\svn@filedir
581   \svnmulti@begininputfilehook
582 }%
```

`\svnmulti@endinputfilehook`

This hook is installed in the `\fink@restore` macro from the `fink` package which will be executed at the end of a input file. The file path `\finkpath` etc. is still valid.

```
583 \def\svnmulti@endinputfilehook{}
584 \message{Package svn-multi: patching macro '\string\fink@restore' from the
585 'fink' package!}%
586 \let\svnmulti@fink@restore\fink@restore
587 \def\fink@restore#1{%
588   \svnmulti@endinputfilehook
589   \svnmulti@fink@restore{#1}%
590   \xdef\svn@filebase{\finkbase}%
591   \xdef\svn@fileext{\finkext}%
592   \xdef\svn@filedir{\finkdir}%
593   \svn@removedotslash\svn@filedir
594 }%
```

`\svnmulti@atbegininputfile`

This macro adds the argument to the end of the `\svnmulti@begininputfilehook`.

```
595 \def\svnmulti@atbegininputfile{%
596   \g@addto@macro\svnmulti@begininputfilehook
597 }
```

`\svnmulti@atendinputfile`

This macro adds the argument to the *begin* of the `\svnmulti@endinputfilehook`. This ensures that code added first is more at the end than code added later. The

code below was adapted from the definition of the L^AT_EX2e macro `\g@addto@macro` which was used above.

```

598 \long\def\svnmulti@endinputfile#1{%
599   \begingroup
600     \@temptokena\expandafter{\svnmulti@endinputfilehook}%
601     \toks@{#1}%
602     \xdef\svnmulti@endinputfilehook{\the\toks@\the\@temptokena}%
603   \endgroup
604 }

605 \def\svn@filestack{[]}
606
607 \def\svn@pushfilestack{%
608   \xdef\svn@filestack{%
609     {\svnfilerev}%
610     {\svnfiledate}%
611     {\svnfileauthor}%
612     {\svnfileyear}%
613     {\svnfilemonth}%
614     {\svnfileday}%
615     {\svnfilehour}%
616     {\svnfileminute}%
617     {\svnfilesecond}%
618     {\svnfiletimezonehour}%
619     {\svnfiletimezoneminute}%
620     {\svnfileurl}%
621     {\svnfilefname}%
622     {\svn@g}%
623     {\svn@pg}%
624   }\svn@filestack}%
625 }
626
627 \def\svn@restorefilekws#1#2\relax{%
628   \svn@restorefilekws@#1\empty
629   \empty \empty \empty \empty
630   \empty \empty \empty \empty
631   \empty \empty \empty \empty
632   \svn@ifempty{#2}%
633     {\gdef\svn@filestack{[]}}%
634     {\gdef\svn@filestack{#2}}%
635 }
636 \def\svn@restorefilekws@#1#2#3#4#5#6#7#8#9{%
637   \gdef\svnfilerev{#1}%
638   \gdef\svnfiledate{#2}%
639   \gdef\svnfileauthor{#3}%
640   \gdef\svnfileyear{#4}%
641   \gdef\svnfilemonth{#5}%
642   \gdef\svnfileday{#6}%
643   \gdef\svnfilehour{#7}%
644   \gdef\svnfileminute{#8}%

```

```

645 \gdef\svnfilesecond{#9}%
646 \svn@restorefilekws@@
647 }
648
649 \def\svn@restorefilekws@@#1#2#3#4#5#6{%
650 \gdef\svnfiletimezonehour{#1}%
651 \gdef\svnfiletimezoneminute{#2}%
652 \gdef\svnfileurl{#3}%
653 \gdef\svnfilename{#4}%
654 \gdef\svn@g{#5}%
655 \gdef\svn@pg{#6}%
656 }
657
658 \def\svn@popfilestack{%
659 \ifx\svn@filestack\empty
660 \PackageWarning{svn-multi}{Underflow of file keyword stack!}%
661 \else
662 \svn@ifequal{\svn@filestack}{{}}%
663 {\PackageWarning{svn-multi}{Underflow of file keyword stack!}}%
664 {\expandafter\svn@restorefilekws\svn@filestack\relax}%
665 \fi
666 }
667
668 \svnmulti@atbegininputfile{%
669 \svn@pushfilestack
670 }
671
672 \svnmulti@atendinputfile{%
673 \svn@popfilestack
674 }%
675 %
676 \fi

```

7.13 Keyword Group Macros

These macros implement the user interface for the keyword group functionality introduced with v2.0.

The list of keyword groups `\svn@glist` is initial set empty and will be filled by `\svngroup`.

```

677 \if@svnmulti@groups
678 \let\svn@glist=\empty

```

`\svngroup`

#1: group name

Saves the group to `\svn@g` and initiates `\svn@g@<group name>@rev` and `\@svn@g@<group name>@rev` if this is the first time the group got used.

The current group symbol ‘*’ is invalid here because there is no way to change to a current group.

```

679 \def\svngroup#1{%
680   \svn@ifequal{#1}{*}%
681   {\PackageError{svn-multi}%
682    {The group name '*' is invalid for '\string\svngroup'}{}{}%
683   }{}%
684   \xdef\svn@g{#1}%
685   \let\svn@pg\svn@g
686   \ifx\svn@g\empty\else%

```

Only initialise the group at first usage:

```

687   \expandafter
688   \ifx\csname @svng@\svn@g @rev\endcsname\relax%
689   \svn@initgroup{\svn@g}%

```

Now save new group to list. The list is checked if its empty to avoid an unwanted leading comma.

```

690   \ifx\svn@glist\empty
691   \xdef\svn@glist{#1}%
692   \else
693   \xdef\svn@glist{\svn@glist,#1}%
694   \fi
695   \fi
696   \fi
697 }

```

\thesvngroup

Returns the current group name to the user.

```

698 \def\thesvngroup{\svn@g}

```

\svnsetcg

#1: group name

Defines `\svn@cg` to the given argument or to `\svn@g` if the argument was `'*`.

```

699 \def\svnsetcg#1{%
700   \svn@ifequal{#1}{*}%
701   {\def\svn@cg{\svn@g}}%
702   {\def\svn@cg{#1}}%
703 }

```

\svncg@def

#1: key name, e.g. `'rev'`, `'date'`

Defines a `\svncgXXX` macro, e.g. `svncgrex`, which returns the requested keyword values of the current keyword group.

```

704 \def\svncg@def#1{%
705   \expandafter
706   \def\csname svncg#1\endcsname{%
707     \@ifundefined{svng@\svn@cg @#1}{??}{}%
708     \csname svng@\svn@cg @#1\endcsname}%
709   }%
710 }

```

```

\svncgrev
\svncgdate
\svncgauthor
\svncgyear
\svncgmonth
\svncgday
\svncghour
\svncgminute
\svncgsecond
\svncgtimezonehour
\svncgtimezoneminute
\svncgurl
\svncgfname

```

Define all \svncgXXX macros by calling \svncg@def in a for loop.

```

711 \@for\@tempa:=%
712   rev,author,date,year,month,day,hour,minute,second,%
713   timezonehour,timezoneminute,url,fname%
714 \do{%
715   \expandafter\svncg@def\expandafter{\@tempa}%
716 }

```

\thesvncg

Simply return the internal macro.

```

717 \def\thesvncg{\svn@cg}

```

\svng

#1: group name
#2: keyword name

Simply returns svng@#1@#2 if defined, '??' otherwise.

```

718 \def\svng#1#2{%
719   \@ifundefined{svng@\svn@temp @#2}%
720   {??}%
721   {\csname svng@\svn@temp @#2\endcsname}%
722 }

```

\svn@addfiletogroup

#1: file name
#2: group name

Adds the given file to the given group. If the group list doesn't exist yet it is initialised. A extra macro for each file is used to remember that the file is already in the group. This could be avoided using a list search.

This is an internal macro so no '*' substitution for the group name.

```

723 \def\svn@addfiletogroup#1#2{%
724   \expandafter
725   \ifx\csname @svg@#2@files@#1\endcsname\relax%
726     \expandafter\gdef\csname @svg@#2@files@#1\endcsname{1}%
727     %
728     \@ifundefined{@svg@#2@files}%
729       {\expandafter\xdef\csname @svg@#2@files\endcsname{#1}}%
730       {\expandafter\xdef\csname @svg@#2@files\endcsname{%
731         \csname @svg@#2@files\endcsname,#1%
732       }}%
733   }%
734 \fi
735 }

```

The input files are added to the list of the current group at their begin to have them before the included graphics and other external files. Special care is taken to not re-initialise the main file which could happen in some special cases (e.g. `\lstinputlisting{\jobname .tex}`).

```

736 \svnmulti@atbegininputfile{%
737   \svn@ifequal{\svn@filepath}{\svn@maindir\jobname.\svn@mainext}%
738   {}%
739   {\svn@initfile}%
740   \svn@ifequal{\svn@fileext}{\svn@mainext}%
741   {\svn@addfiletogroup{\svn@filedir\svn@filebase}{\svn@pg}}}%
742   \svn@ifequal{\svn@fileext}{sty}%
743   {\svn@addfiletogroup{\svn@filedir\svn@filebase}{\svn@pg}}}%
744   \svn@ifequal{\svn@fileext}{cls}%
745   {\svn@addfiletogroup{\svn@filedir\svn@filebase}{\svn@pg}}}%
746   \svn@addfiletogroup{\svn@filepath}{\svn@filedir\svn@filebase}%
747 }

```

\svn@writegroup

#1: group name

Writes group to \svn@write file.

```

748 \def\svn@writegroup#1{%
749   \def\svn@writekw##1{%
750     \immediate\write\svn@write{%
751       \noexpand\@namedef{svg@#1@##1}{\csname @svg@#1@##1\endcsname}%
752     }%
753   }%
754   \svn@writekw{rev}%
755   \svn@writekw{date}%
756   \svn@writekw{author}%
757   \svn@writekw{year}%
758   \svn@writekw{month}%
759   \svn@writekw{day}%
760   \svn@writekw{hour}%
761   \svn@writekw{minute}%
762   \svn@writekw{second}%

```

```

763 \svn@writekw{timezonehour}%
764 \svn@writekw{timezoneminute}%
765 \@ifundefined{@svg@#1@files}{-}{%
766 \immediate\write\svn@write{%
767 \noexpand
768 \svn@namegdefverb{svg@#1@files}{\csname @svg@#1@files\endcsname}%
769 }%
770 }%
771 \immediate\write\svn@write{%
772 \noexpand
773 \svn@namegdefverb{svg@#1@url}{\csname @svg@#1@url\endcsname}^^J%
774 \noexpand
775 \svn@namegdefverb{svg@#1@fname}{\csname @svg@#1@fname\endcsname}^^J%
776 }%
777 }

```

\svn@writeallgroups

#1: macro holding a list of groups

```

778 \def\svn@writeallgroups#1{%
779 \begingroup
780 \ifx\relax#1\relax\else
781 \@for\svn@temp:=#1\do{%
782 \svn@ifvalidrev{@svg@\svn@temp @rev}%
783 {%
784 \expandafter
785 \svn@cleanfilelist\csname @svg@\svn@temp @files\endcsname
786 \svn@writegroup{\svn@temp}%
787 \@ifundefined{@svg@\svn@temp @files}{-}{%
788 {\expandafter\svn@writeallgroups
789 \csname @svg@\svn@temp @files\endcsname
790 }%
791 }-}%
792 }%
793 \fi
794 \endgroup
795 }

```

\svn@updategroup

#1: group name

Updates group with \svnfile... macro values.

```

796 \def\svn@updategroup#1{%
797 \@ifundefined{@svg@#1@rev}%
798 {\svn@initgroup{#1}}%
799 }%
800 \expandafter
801 \ifnum\csname @svg@#1@rev\endcsname<\svnfilerev
802 \svn@gkwset{#1}{rev}{\svnfilerev}%
803 \svn@gkwset{#1}{date}{\svnfiledate}%

```

```

804 \svn@gkwset{#1}{author}{\svnfileauthor}%
805 \svn@gkwset{#1}{year}{\svnfileyear}%
806 \svn@gkwset{#1}{month}{\svnfilemonth}%
807 \svn@gkwset{#1}{day}{\svnfileday}%
808 \svn@gkwset{#1}{hour}{\svnfilehour}%
809 \svn@gkwset{#1}{minute}{\svnfileminute}%
810 \svn@gkwset{#1}{second}{\svnfilesecond}%
811 \svn@gkwset{#1}{timezonehour}{\svnfiletimezonehour}%
812 \svn@gkwset{#1}{timezoneminute}{\svnfiletimezoneminute}%
813 \svn@gkwset{#1}{url}{\svnfileurl}%
814 \svn@gkwset{#1}{fname}{\svnfilefname}%
815 \fi
816 }

```

\svn@definegroup

#1: group name

Defines group value so that they are available for the user, e.g. instead of the internal @svng@... macros it sets the svng@... macros. This is done by calling **\svn@updategroup** with a modified version of **\svn@gkwset**.

```

817 \def\svn@definegroup#1{%
818 \svn@gkwdef{#1}{rev}%
819 \svn@gkwdef{#1}{date}%
820 \svn@gkwdef{#1}{author}%
821 \svn@gkwdef{#1}{year}%
822 \svn@gkwdef{#1}{month}%
823 \svn@gkwdef{#1}{day}%
824 \svn@gkwdef{#1}{hour}%
825 \svn@gkwdef{#1}{minute}%
826 \svn@gkwdef{#1}{second}%
827 \svn@gkwdef{#1}{timezonehour}%
828 \svn@gkwdef{#1}{timezoneminute}%
829 \svn@gkwdef{#1}{url}%
830 \svn@gkwdef{#1}{fname}%
831 }

```

\svn@initgroup

#1: group name

Initialises group.

```

832 \def\svn@initgroup#1{%
833 \svn@gkwset{#1}{rev}{-1}%
834 \svn@gkwset{#1}{date}{}%
835 \svn@gkwset{#1}{author}{}%
836 \svn@gkwset{#1}{year}{0000}%
837 \svn@gkwset{#1}{month}{00}%
838 \svn@gkwset{#1}{day}{00}%
839 \svn@gkwset{#1}{hour}{00}%
840 \svn@gkwset{#1}{minute}{00}%
841 \svn@gkwset{#1}{second}{00}%

```



```

842 \svn@gkwset{#1}{timezonehour}{+00}%
843 \svn@gkwset{#1}{timezoneminute}{00}%
844 \svn@gkwset{#1}{url}{}%
845 \svn@gkwset{#1}{fname}{}%
846 }

```

\svn@gkwset

#1: group name
#2: keyword name
#3: value

Sets *<value>* for *<keyword>* in *<group>*.

```

847 \def\svn@gkwset#1#2#3{%
848   \expandafter
849   \xdef\csname @svng@#1@#2\endcsname{#3}%
850 }

```

\svn@gkwdef

#1: group name
#2: keyword name

Defines `svng@...` macros used by the user macros to the value of the internal `@svng@...` macros.

```

851 \def\svn@gkwdef#1#2{%
852   \expandafter
853   \xdef\csname svng@#1@#2\endcsname{\csname @svng@#1@#2\endcsname}%
854 }

```

\svn@cleanfilelist

#1: macro holding a file list

Takes a macro which holds a file list and removes all files from the list which don't have a valid revision number.

```

855 \def\svn@cleanfilelist#1{
856   \begingroup
857   \def\svn@tmplist{}%
858   \ifx\relax#1\relax\else
859     \@for\svn@temp:=#1\do{%
860       \expandafter\svn@ifvalidrev
861       \expandafter{\svng@\svn@temp @rev}%
862       {\edef\svn@tmplist{\svn@tmplist,\svn@temp}}%
863     }%
864   }%
865   \xdef#1{\expandafter\@gobble\svn@tmplist\empty}%
866   \fi
867 \endgroup
868 }
869 \fi

```

7.14 Files as extra groups

Macros which allow single files to be declared as extra groups so that their keywords can be accessed in the whole document like with normal groups. This special groups are not added to the list of groups.

A user-level switch is declared to enable or disable the automatic declaration of every file as own group. This causes `\svnsubgroup` to be called for all input files. The if macro is defined outside the `\if@svnmulti@subgroups` because `\newif` inside `\if` is not a good idea.

```
870 \newif\ifsvnsubgroups
871 \svnsubgroupsfalse
872 \if@svnmulti@subgroups
873 \svnsubgroupstrue
```

`\svnsubgroup`

User level and internal macro to declare the current file as extra group. It produces the current file path and calls `\svn@subgroup`. Creates two groups one with and one without the file extension. The one without holds the latest revision of all files included in this file.

```
874 \def\svnsubgroup{%
875   \begingroup
876   \svn@removedotslash\svn@filedir
877   \svn@subgroup{\svn@filedir\svn@filebase}%
878   \svn@subgroup{\svn@filepath}%
879   \endgroup
880 }
```

`\svn@subgroup`

#1: file name

Macro to write a file as group to `.svn` file. After checking if the filename was not already written, the `.svn` file is checked if it is open and then the file keyword information is written.

```
881 \def\svn@subgroup#1{%
882   \ifnum\svnfilerev>-1\relax
883   \expandafter\ifx\csname svn@g@#1\endcsname\relax%
884   \expandafter\gdef\csname svn@g@#1\endcsname{1}%
885   \svn@updategroup{#1}%
886   \fi
887   \fi
888 }
```

`\svnignoreextensions`

#1: A comma separated list of file name extensions (without leading dot) to ignore for automatic `\svnsubgroup`.

A special macro is defined for all extensions. The existents of this macro is then tested later to check if this extension should be ignored.

```

889 \def\svnignoreextensions#1{%
890   \@for\svn@temp:=#1\do{%
891     \expandafter\def\csname svn@ignore@ext@\svn@temp\endcsname{}%
892   }%
893 }

```

\svnconsiderextensions

#1: A comma separated list of file name extensions (without leading dot) to consider for automatic **\svnsubgroup**.

The special macro defined by **\svnignoreextensions** is deleted, i.e. **\let** to **\relax**.

```

894 \def\svnconsiderextensions#1{%
895   \@for\svn@temp:=#1\do{%
896     \expandafter\let\csname svn@ignore@ext@\svn@temp\endcsname\relax%
897   }%
898 }

```

The following extensions are ignored by default.

```

899 \svnignoreextensions{aux,bbl,cfg,cls,fd,enc,fls,glo,idx,ilg,ind,ist,%
900 lof,log,lot,out,sty,svn,svt,svx,toc}

```

Check at the end of every input file if files should be extra groups and declare this file as group if its extension is not configured to be ignored.

```

901 \svnmulti@atendinginputfile{%
902   \if@svnmulti@subgroups
903     \ifsvnsubgroups
904       \expandafter\ifx\csname svn@ignore@ext@\svn@fileext\endcsname\relax
905       \svnsubgroup
906     \fi
907   \fi
908 \fi
909 }

910 \AtBeginDocument{%
911   \if@svnmulti@subgroups
912     \ifsvnsubgroups
913       \svn@addfiletogroup{\jobname .\svn@mainext}{\jobname}%
914     \svnsubgroup
915   \fi
916 \fi
917 }

918 \fi

```

7.15 External Files

Macros to declare external files and load the keywords from **.svx** files generated by **svn-multi.pl**.

```

919 \if@svnmulti@external

```

`\svnexternalgroup`

#1: group name

Defines the default group of external files. The default is to always use the current group. An empty argument puts the external files in no group. A ‘*’ switches back to always use the current group.

```
920 \if@svnmulti@groups
921 \def\svnexternalgroup#1{%
922   \svn@ifequal{#1}{*}%
923     {\def\svn@externalgroup{#1}}%
924     {\def\svn@externalgroup{\svn@g}}%
925 }
926 \def\svn@externalgroup{\svn@g}
927 \else
928 \def\svn@externalgroup{}
929 \fi
```

`\svnexternal`

#1: group name

#2: list of filenames in { }

Writes the current input file path and its argument as arguments of `\@svnexternal` into the `.svn` file.

```
930 \newcommand*\svnexternal[2][]{%
931   \if@filesw
932     \svn@checkwrite
933     \begingroup
934       \svn@ifequal{#1}{*}%
935         {\def\svn@temp{\svn@g}}%
936         {\svn@ifempty{#1}%
937           {\def\svn@temp{\svn@externalgroup}}%
938           {\def\svn@temp{#1}}%
939         }%
940       \immediate\write\svn@write{%
941         \noexpand\@svnexternal[\svn@temp]{\svn@filepath}{#2}%
942       }%
943     \endgroup
944   \fi
945   \svn@inputsvx{\svn@filedir\svn@filebase}%
946 }
```

`\svnexternalpath`

#1: list of paths in { }

Writes its argument as argument of `\@svnexternalpath` into the `.svn` file.

```
947 \def\svnexternalpath#1{%
948   \if@filesw
949     \svn@checkwrite
950     \immediate\write\svn@write{%
951       \noexpand\@svnexternalpath{#1}%
952     }%
953   \fi
954 }
```

```

952     }%
953   \fi
954 }

```

\@svnexternal

\@svnexternalpath

Discards the argument(s). These macros and their arguments are only used by the external `svn-multi.pl` script.

```

955 \newcommand*\@svnexternal[3][]{ }
956 \def\@svnexternalpath#1{ }

```

\svnexternalfile

This macro is generated by `svn-multi.pl` and should not be used by the user. If files-as-group is enabled some special characters are disabled and the `\svn@externalfile` is called to read the file name. Otherwise the argument is simply removed.

```

957 \newcommand*\svnexternalfile[1][\svn@filedir\svn@filebase]{%
958   \begin{group} % TODO: maybe use \svn@catcodes
959     \catcode'\_ =12
960     \catcode'\& =12
961     \catcode'\^ =12
962     \catcode'\$ =12
963     \catcode'\# =12
964     \svn@externalfile{#1}%
965 }

```

\svn@externalfile

#1: group name

#2: file name

Ends group which began in `\svnexternalfile` and calls the appropriate macros.

```

966 \def\svn@externalfile#1#2{%
967   \endgroup
968   \if@svnmulti@subgroups
969     \ifsvnsubgroups
970       \svn@ifequal{#1}{\svn@rg}%
971       {\svn@addfiletogroup{#2}{\svn@filedir\svn@filebase}}%
972       {\svn@addfiletogroup{#2}{#1}}%
973       \svn@subgroup{#2}%
974   \fi
975 \fi
976 }

```

If **external** option is not enabled a placeholder macro is defined which simply ignores its argument.

```

977 \else
978   \def\svnexternalfile#1{%
979 \fi

```

7.16 Auto loading of .svx files

Auto loading of .svx files at the begin of `\input` or `\include` files using the `\svnmulti@atbegininputfile` macro. The macros `\svn@addfiletogroup` and `\svnsubgroup` are used to do the actual work.

```
980 \if@svnmulti@autoload
981
982 \svnmulti@atbegininputfile{%
983   \svn@ifequal{\svn@fileext}{tex}%
984     {\svn@inputsvx{\svn@filedir\svn@filebase}}%
985   }%
986 }
```

The main .svx is loaded at the end of the package.

```
987 %%\AtEndOfPackage{%
988 \AtBeginDocument{%
989   \svn@inputsvx{\jobname}%
990 }
991 \fi
```

7.17 Support for Graphic Packages

7.17.1 Common Code

```
992 \if@svnmulti@anygraphic
```

`\svngraphicsgroup`

#1: graphic group name

Defines the default group of graphics files. The default is empty which means the current group.

```
993 \def\svngraphicsgroup#1{%
994   \svn@ifequal{#1}{*}%
995     {\def\svn@graphicsgroup{\svn@g}}%
996     {\def\svn@graphicsgroup{#1}}%
997 }
998 \def\svn@graphicsgroup{\svn@externalgroup}
```

`\svnignoregraphic`

#1: file name/path

Ignores the given graphic file by defining a special macro.

```
999 \def\svnignoregraphic#1{%
1000   \expandafter\def\csname svn@ignoregraphic@#1\endcsname{%
1001 }
```

`\svnconsidergraphic`

#1: file name/path

Deletes the special ignore macro to consider the graphic again.

```
1002 \def\svnconsidergraphic#1{%
```

```

1003 \expandafter\let\csname svn@ignoregraphic@#1\endcsname\relax%
1004 }
1005 \fi

```

7.17.2 Package graphics

Automatic declaration of all images included by `\includegraphics` from the `graphics` package as external files. We use the `\Gin@setfile` macro from that package which receives the image file name as third argument.

```

1006 \if@svnmulti@graphics
1007 \RequirePackage{graphics}[2006/02/20]

```

`\Gin@setfile`

```

#1: ??, not used
#2: ??, not used
#3: graphic file name/path

1008 \message{Package svn-multi: patching macro '\string\Gin@setfile' from the
1009 'graphics' package!}%
1010 \let\svnmulti@Gin@setfile\Gin@setfile
1011 \renewcommand*{\Gin@setfile}[3]{%
1012   \expandafter\ifx\csname svn@ignoregraphic@#3\endcsname\relax%
1013     \svnexternal[\svn@graphicsgroup]{\{#3\}}%
1014   \fi
1015   \svnmulti@Gin@setfile{#1}{#2}{#3}%
1016 }
1017 \fi

```

7.17.3 Package pgf

The `pgf` macro `\pgf@declareimage` which is called by the user macro `\pgfdeclareimage` is used.

```

1018 \if@svnmulti@pgfimages
1019 \RequirePackage{pgf}[2008/01/15]

```

`\pgf@declareimage`

```

#1: ??, not used
#2: image label
#3: ??, not used

1020 \message{Package svn-multi: patching macro '\string\pgf@declareimage' and will
1021 patch generated macros '\string\pgf@image@<name>!' from the 'pgf' package!}%
1022 \let\svnmulti@pgf@declareimage\pgf@declareimage
1023 \renewcommand*{\pgf@declareimage}[3][{}]{%
1024   \svnmulti@pgf@declareimage[#1]{#2}{#3}%

```

At this point the used image filename is defined by `\pgf@filename` and the image itself is defined by `\pgf@image@#2!` which is a `\let` copy of temporary `\pgf@image`. An own copy of this is created and the old name `\pgf@image@#2!`

is used to execute `\svnexternal` every time the image is included using `\pgfuseimage`.

```

1025 \ifx\pgf@filename\empty\else
1026 \expandafter\ifx\csname svn@ignoregraphic@\pgf@filename\endcsname\relax%
1027 \expandafter\global\expandafter%
1028 \let\csname svnmulti@pgf@image@#2!\endcsname=\pgf@image%
1029 \expandafter\xdef\csname pgf@image@#2!\endcsname{%
1030 \noexpand\svnexternal[\noexpand\svn@graphicsgroup]{\pgf@filename}}%
1031 \csname svnmulti@pgf@image@#2!\endcsname
1032 }%
1033 \fi
1034 \fi
1035 }
1036 \fi

```

7.18 Table of Revisions

```

1037 \if@svnmulti@table
1038 \ifx\tableofcontents\relax\else

```

`\svnrevisionsname`

Simple definition for now. Language support over ‘babel’s `\language` possible.

```

1039 \def\svnrevisionsname{Table of Revisions}%

```

`\svn@svt`

File ending for table of revision auxiliary file. A macro is used to allow redefinition by the user if another package is uses the same ending.

```

1040 \def\svn@svt{svt}

```

`\tableofrevisions`

The `\tableofcontents` macro from standard L^AT_EX is adapted for this macro. Classes which provide chapters will get a different table then one which not.

```

1041 \AtBeginDocument{%
1042 \ifx\chapter\@undefined
1043
1044 %% Adapted from the \tableofcontents macro, LaTeX ‘article’ class [2005/09/16]
1045 \newcommand\tableofrevisions{%
1046 \section*{\svnrevisionsname
1047 \@mkboth{\MakeUppercase\svnrevisionsname}{\MakeUppercase\svnrevisionsname}}%
1048 \svn@input{\jobname .\svn@svt}%
1049 }
1050
1051 \else
1052
1053 %% Adapted from the \tableofcontents macro, LaTeX ‘book’ class [2005/09/16]
1054 \newcommand\tableofrevisions{%
1055 \if@twocolumn

```



```

1056 \@restonecoltrue\onecolumn
1057 \else
1058 \@restonecolfalse
1059 \fi
1060 \chapter*{\svnrevisionsname
1061 \mkboth{\MakeUppercase\svnrevisionsname}{\MakeUppercase\svnrevisionsname}}%
1062 \svn@input{\jobname .\svn@svt}%
1063 \if@restonecol\twocolumn\fi
1064 }
1065
1066 \fi
1067 }
1068 \fi % defined \tableofcontents

```

\svn@writerow

#1: row type ('group', 'file', 'global', ...)
#2: row type specific argument
#3: row type specific argument

Writes a table row by using \svn@tabcell and \svn@tabcellarg defined by the \svn@writeXXXrow macro below.

```

1069 \def\svn@writerow#1#2#3{%
1070 \immediate\write\svn@svtwrite{%
1071 \expandafter\noexpand\csname svn#1row\endcsname
1072 \expandafter\noexpand\csname svntab#1\endcsname{#2}{#3}\space
1073 \@ampersamchar\space
1074 \svn@tabcell{rev}\space\@ampersamchar\space
1075 \svn@tabcell{author}\space\@ampersamchar\space
1076 \noexpand\svntabdate%
1077 \svn@tabcellarg{year}%
1078 \svn@tabcellarg{month}%
1079 \svn@tabcellarg{day}%
1080 \svn@tabcellarg{hour}%
1081 \svn@tabcellarg{minute}%
1082 \svn@tabcellarg{second}%
1083 \svn@tabcellarg{timezonehour}%
1084 \svn@tabcellarg{timezoneminute}%
1085 \space\@backslashchar\@backslashchar
1086 \expandafter\noexpand\csname endsvn#1row\endcsname
1087 }%
1088 }

```

\svn@writegrouprow

#1: current group

```

1089 \def\svn@writegrouprow#1{%
1090 \begingroup
1091 \def\svn@tabcellarg##1{\csname @svng@#1##1\endcsname}}%
1092 \def\svn@tabcell##1{\expandafter\noexpand\csname svntab##1\endcsname%
1093 \svn@tabcellarg{##1}%

```

```

1094 }%
1095 \svn@writerow{group}{#1}{}%
1096 \endgroup
1097 }

```

\svn@writesubgrouprow

#1: grouping level
#2: subgroup name

```

1098 \def\svn@writesubgrouprow#1#2{%
1099 \begingroup
1100 \def\svn@tabcellarg##1{\csname @svng@#2@##1\endcsname}}%
1101 \def\svn@tabcell##1{\expandafter\noexpand\csname svntab##1\endcsname}%
1102 \svn@tabcellarg{##1}%
1103 }%
1104 \svn@writerow{subgroup}{#1}{#2}%
1105 \endgroup
1106 }

```

\svn@writefilerow

#1: grouping level
#2: file name

```

1107 \def\svn@writefilerow#1#2{%
1108 \begingroup
1109 \def\svn@tabcellarg##1{\csname @svng@#2@##1\endcsname}}%
1110 \def\svn@tabcell##1{\expandafter\noexpand\csname svntab##1\endcsname}%
1111 \svn@tabcellarg{##1}%
1112 }%
1113 \svn@writerow{file}{#1}{#2}%
1114 \endgroup
1115 }

```

\svn@writeglobalrow

```

1116 \def\svn@writeglobalrow{%
1117 \begingroup
1118 \def\svn@tabcellarg##1{\csname @svn@##1\endcsname}}%
1119 \def\svn@tabcell##1{\expandafter\noexpand\csname svntab##1\endcsname}%
1120 \svn@tabcellarg{##1}%
1121 }%
1122 \svn@writerow{global}{}{}%
1123 \endgroup
1124 }

```

7.18.1 Table Format Macros

Generic format macro used in the .svt file. Can be redefined by the user to change table format. TODO: More documentation needed!

\svntable

```

1125 \def\svntable{%
1126 \begin{tabular}{p{0.425\textwidth}rp{0.25\textwidth}p{0.225\textwidth}}%

```

```

1127     \hline
1128 }

\endsvntable

1129 \def\endsvntable{\hline\end{tabular}}

\svntablehead

1130 \def\svntablehead{%
1131     Name & Rev & Last Author & Last Changed At \\ \hline
1132 }

\svntablefoot

1133 \def\svntablefoot{}

\svnbeforetable

1134 \def\svnbeforetable{}

\svnaftertable

1135 \def\svnaftertable{\clearpage}

\svnglobalrow

1136 \def\svnglobalrow{}

\endsvnglobalrow

1137 \def\endsvnglobalrow{}

\svngrouprow

1138 \def\svngrouprow{}

\endsvngrouprow

1139 \def\endsvngrouprow{}

\svnsubgrouprow

1140 \def\svnsubgrouprow{}

\endsvnsubgrouprow

1141 \def\endsvnsubgrouprow{}

\svnsubgrouprow

1142 \def\svnfilerow{}

\endsvnfilerow

1143 \def\endsvnfilerow{}

```

\svntabglobal

```
1144 \def\svntabglobal{Document}
```

\svntabgroup

```
1145 \def\svntabgroup#1{Group '#1'}
```

\svntabfile

```
1146 \def\svntabsubgroup#1{%
1147   \raggedright
1148   \addtolength{\leftskip}{#1\medskipamount}%
1149   \begingroup
1150   \catcode'\_ =12
1151   \catcode'\& =12
1152   \catcode'\^ =12
1153   \catcode'\$ =12
1154   \catcode'\# =12
1155   \svn@tabsubgroup
1156 }
1157 \def\svn@tabsubgroup#1{\endgroup Subgroup '\texttt{\small #1}'}
```

\svntabfile

```
1158 \def\svntabfile#1{%
1159   \raggedright
1160   \addtolength{\leftskip}{#1\medskipamount}%
1161   \begingroup
1162   \catcode'\_ =12
1163   \catcode'\& =12
1164   \catcode'\^ =12
1165   \catcode'\$ =12
1166   \catcode'\# =12
1167   \svn@tabfile
1168 }
1169 \def\svn@tabfile#1{\endgroup File '\texttt{\small #1}'}
```

\svntabrev

```
1170 \def\svntabrev{}
```

\svntabauthor

```
1171 \def\svntabauthor#1{\svnFullAuthor{#1}}
```

\svntabdate

```
1172 \def\svntabdate#1#2#3#4#5#6#7#8{%
1173   #1-#2-#3 #4:#5:#6% #7#8%
1174 }
1175 \fi
```

7.19 Other macros

This section contains macros which don't fit in any other section.

\svn

\svn*

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

```
1176 \newcommand{\svn}{\@ifnextchar{*}{\svn@s}{\svn@n}}
1177 \def\svn@n#1{\@svn@n#1}
1178 \def\svn@s*#1{\@svn@s#1}
1179 \def\@svn@n$#1$#1}
1180 \def\@svn@s$#1 $#1}
```

\svnnolinkurl

#1: URL

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.

```
1181 %% Adapted from the \url macro of the 'hyperref' package.
1182 \DeclareRobustCommand*\svnnolinkurl{%
1183   \@ifundefined{hyper@normalise}%
1184     {\PackageWarning{svn-multi}{Package hyperref is needed for \noexpand
1185       \svnnolinkurl.}}%
1186     {\hyper@normalise\svnnolinkurl@}%
1187 }%
1188 \def\svnnolinkurl@#1{\Hurl{#1}}%
```

\svn@getfilename

#1: URL

This macro expands the content using the temporary macro and sets it in front of the `\svn@getfilename` sub-macro together with `/{}` to make sure the macro does not break at values without directories. A `\relax` is used as end marker.

```
1189 \def\svn@getfilename#1{%
1190   \begingroup
1191     \edef\svn@temp{#1}%
1192     \expandafter\@svn@getfilename\svn@temp/{ }\relax
1193 }%
```

\@svn@getfilename

#1: URL part before first slash

#2: URL part after first slash

Splits the content at the first slash (`/`) and checks if the remainder is empty. If so the end marker got reached and the part before the slash is the filename which is returned. Otherwise the macro recursively calls itself to split the remainder.

```

1194 \def\@svn@getfilename#1/#2\relax{%
1195     \svn@ifempty{#2}%
1196     {\endgroup\gdef\svnfilename{#1}}%
1197     {\@svn@getfilename#2\relax}%
1198 }%

```

7.20 Auxiliary file generation and read-back

Reread output from last compile run if it exists.

```

1199 \@input{\jobname .svn}

```

\svn@checkwrite

Checks if .svn file is already open and if not opens it. This makes sure that the file is only created if really needed. The macro is only needed once, so it's redefines itself to `\relax` at the end.

```

1200 \def\svn@checkwrite{%
1201     \@ifundefined{svn@write}{%
1202         \newwrite\svn@write
1203         \immediate\openout\svn@write=\jobname.svn\relax%
1204         \immediate\write\svn@write{\@percentchar\space SVN Keyword cache}%
1205         %\immediate\write\svn@write{\noexpand\makeatletter}%
1206     }{}%
1207     \let\svn@checkwrite=\relax
1208 }

```

\svn@writesvn

This macro writes the .svn auxiliary file and is called from a `\AtEndDocument` macro later on.

```

1209 {\catcode'\&=12
1210 \gdef\@ampersamchar{&}
1211 }
1212 \def\svn@writesvn{%

```

Remove all files which do not have a revision number from list:

```

1213     \if@svnmulti@groups
1214     \fi

```

Write document global values:

```

1215     \svn@checkwrite
1216     \immediate\write\svn@write{^^J%
1217         \@percentchar\space Global values:^^J%
1218         \noexpand\def\noexpand\svnrev{\@svn@rev}^^J%
1219         \noexpand\def\noexpand\svndate{\@svn@date}^^J%
1220         \noexpand\def\noexpand\svnauthor{\@svn@author}^^J%
1221         \noexpand\def\noexpand\svnyear{\@svn@year}^^J%
1222         \noexpand\def\noexpand\svnmonth{\@svn@month}^^J%
1223         \noexpand\def\noexpand\svnday{\@svn@day}^^J%
1224         \noexpand\def\noexpand\svnhour{\@svn@hour}^^J%
1225         \noexpand\def\noexpand\svnminute{\@svn@minute}^^J%

```

```

1226 \noexpand\def\noexpand\svnsecond{\@svn@second}^^J%
1227 \noexpand\def\noexpand\svntimezonehour{\@svn@timezonehour}^^J%
1228 \noexpand\def\noexpand\svntimezoneminute{\@svn@timezoneminute}^^J%
1229 \noexpand\svn@gdefverb\noexpand\svnurl{\@svn@url}^^J%
1230 \noexpand\svn@gdefverb\noexpand\svnfname{\@svn@fname}^^J%
1231 }%

```

Write group keyword macro definitions. Remove all files which do not have a revision number from list:

```

1232 \if@svnmulti@groups
1233 \svn@cleanfilelist\@svng@@files
1234 \immediate\write\svn@write{%
1235 \noexpand\def\noexpand\svng@@files{\@svng@@files}^^J%
1236 }%
1237 \svn@writeallgroups\@svng@@files

```

Write keyword group values if groups were specified:

```

1238 \ifx\svn@glist\empty\else
1239 \begingroup
1240 \immediate\write\svn@write{^^J%
1241 \percentchar\space SVN File Groups: \svn@glist
1242 }%
1243 \svn@writeallgroups\svn@glist
1244 \endgroup
1245 \fi
1246 \else
1247 \immediate\write\svn@write{^^J}%
1248 \fi

```

Finally close output file:

```

1249 \immediate\closeout\svn@write%
1250 }

```

\svn@writegroupfiles

#1: group name

Writes all files of a group to the .svt file. If the file is actually a subgroup it calls itself recursively.

```

1251 \def\svn@writegroupfiles#1{%
1252 \begingroup
1253 \advance\svn@grouplevel by 1\relax
1254 \expandafter\let
1255 \expandafter\svn@files\csname @svng@#1@files\endcsname
1256 \ifx\svn@files\relax\else
1257 \svn@cleanfilelist\svn@files
1258 \@for\svn@file:=\svn@files\do{%
1259 \svn@ifvalidrev{\svng@\svn@file @rev}%
1260 {%
1261 \@ifundefined{\svng@\svn@file @files}%
1262 {\svn@writefilerow{\the\svn@grouplevel}{\svn@file}}%
1263 {\svn@ifequal{\csname @svng@\svn@file @files\endcsname}{\svn@file.tex}%

```

```

1264             {\svn@writefilerow{\the\svn@grouplevel}{\svn@file.tex}}%
1265             {%
1266             \svn@writesubgrouprow{\the\svn@grouplevel}{\svn@file}%
1267             \svn@writegroupfiles{\svn@file}%
1268             }%
1269         }%
1270     }{}%
1271 }%
1272 \fi
1273 \endgroup
1274 }%

```

\svn@writesvt

This macro writes the .svt auxiliary file and is called from a \AtEndDocument macro later on.

```

1275 \def\svn@writesvt{%
    Write table of revisions if enabled.
1276 \if@svnmulti@table
    Open .svt file and write table header:
1277 \newwrite\svn@svtwrite
1278 \immediate\openout\svn@svtwrite=\jobname.\svn@svt\relax
1279 \@onelevel@sanitize\svntable%
1280 \immediate\write\svn@svtwrite{%
1281     \noexpand\svnbeforetable^^J%
1282     \svntable
1283     \noexpand\svntablehead^^J%
1284 }%

```

Group rows:

```

1285 \let\svn@grouplevel\@tempcnta
1286 \svn@grouplevel=0\relax
1287 \svn@writeglobalrow{}%
1288 \svn@writegroupfiles{}%
1289 %
1290 \@for\svn@g:=\svn@glist\do{%
1291     \@ifundefined{@svng@\svn@g @rev}{}%
1292     {%
1293         \expandafter
1294         \ifnum\csname @svng@\svn@g @rev\endcsname>0\relax
1295             \svn@writegrouprow{\svn@g}%
1296             \svn@writegroupfiles{\svn@g}%
1297         \fi
1298     }%
1299 }%
1300

```

Write table footer and close file:

```

1301 \@onelevel@sanitize\endsvntable%
1302 \immediate\write\svn@svtwrite{%

```



```

1303      \noexpand\svntablefoot^^J%
1304      \endsvntable^^J%
1305      \noexpand\svnaftertable
1306    }%
1307    \immediate\closeout\svn@svtwrite%
1308  \fi
1309 }

```

At the end of document the values are written to the auxiliary file.

```

1310 \AtEndDocument{%
1311   \if@filesw
1312     \ifx\@svn@rev\empty\else
1313       \ifnum\@svn@rev<1\else
1314         \svn@writesvn
1315         \svn@writesvt
1316       \fi
1317     \fi
1318   \fi
1319 }

```

7.21 Backward compatibility wrapper `svnkw.sty`

For backward compatibility a wrapper file with the old package name `svnkw` is provided. Newer documents should use the name `svn-multi`.

```

1 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
2 \ProvidesPackage{svnkw}
3 [\filedate\space\fileversion\space Backward compatibility wrapper for svn-multi]
4 \PackageWarning{svnkw}{The package 'svnkw' got renamed to 'svn-multi'
5 and is now only a backward compatibility wrapper which loads 'svn-multi'.
6 Please adjust your document preamble to use the new name.}
7 \RequirePackage{svn-multi}[\filedate]

```

Change History

v1.0		Added warning when a wrong, maybe misspelled, keyword is given.	1 34
v1.1	General: Initial version		1
	General: Added macros to extract and typeset date/time information. Added macros to set and typeset main URL or filename.	Macro now returns the username if the full name was not registered.	1 38
v1.2		New macro	37, 39, 61
	General: Added star version. Normal version was not changed to not break existing documents with user defined keywords without leading space.	Renamed package from <code>svnkw</code> to <code>svn-multi</code> to match CTAN directory. Wrapper file <code>svnkw.sty</code> is provided for backward compatibility.	61 1

v1.3	General: Added verbatim support. Keywords can now contain special character like <code>_ ^ \$ % & \</code> . Rewrote keyword check macros to work with verbatim code. <code>\nofiles</code> is now obeyed. 1	which modify the list of special characters. 1
v1.3a	General: Fixed issue with unwanted spaces generated by <code>\svnid</code> , <code>\svnidlong</code> and <code>\svnkwsave</code> , e.g. when used in a file which is included with <code>\input</code> 1	v1.4 General: Added support for timezones with non-zero minute part, e.g. <code>+0530</code> 1 Return now full timezone (hour + minute part). Manually added 00 minutes are removed. 26
v1.3b	General: Changed the way catcodes are modified to be compatible with the french option of the babel package or other packages	v1.5 General: Added <code>\today</code> -style macros <code>\svntoday</code> and <code>\svnfiletoday</code> 1
		v2.0 General: New features: keyword groups, external file support, auto-include of images, files as subgroups and table of revisions. 1

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		
<code>\#</code> 963, 1154, 1166	<code>\@svn@month</code> 172, 265, 1222	<code>\@svn@updateid</code> 247, 253, 338
<code>\\$</code> 361, 368, 962, 1153, 1165	<code>\@svn@n</code> 1177, 1179	<code>\@svn@url</code> ... 272, 1229
<code>\&</code> . 960, 1151, 1163, 1209	<code>\@svn@parsetimezone</code> 475, 477	<code>\@svn@year</code> 171, 264, 1221
<code>\@input</code> 155	<code>\@svn@rev</code> 168, 260, 261, 1218, 1312, 1313	<code>\@svncg@save</code> 289
<code>\@input</code> 1199	<code>\@svn@es</code> 1178, 1180	<code>\@svnexternal</code> . 941, 955
<code>\@nameundef</code> 164	<code>\@svn@scandate</code> 246, 454	<code>\@svnexternalpath</code> 951, 956
<code>\@onelevel@sanitize</code> 1279, 1301	<code>\@svn@scandate@</code> 454, 456	<code>\@svng@@files</code> . 184, 1233, 1235, 1237
<code>\@svn@author</code> 170, 263, 1220	<code>\@svn@scanlongdate</code> 336, 466	<code>\@svnidswfalse</code> 244, 354
<code>\@svn@date</code> 169, 262, 1219	<code>\@svn@scanlongdate@</code> 466, 468	<code>\@svnidswtrue</code> 240
<code>\@svn@day</code> 173, 266, 1223	<code>\@svn@second</code> 176, 269, 1226	<code>\@svnmulti@anygraphictrue</code> 104, 107
<code>\@svn@fname</code> .. 273, 1230	<code>\@svn@timezonehour</code> 177, 270, 1227	<code>\@svnmulti@autoload</code> 82, 84, 87, 93, 100
<code>\@svn@getfilename</code> 1192, 1194, 1197	<code>\@svn@timezoneminute</code> ... 178, 271, 1228	<code>\@svnmulti@autoloadfalse</code> 19, 97
<code>\@svn@hour</code> 174, 267, 1224		<code>\@svnmulti@autoloadtrue</code> 30, 88
<code>\@svn@minute</code> 175, 268, 1225		

\@svnmulti@externalfalse	F	N
..... 16	\fink@file 564	\next 552, 553, 554
\@svnmulti@externaltrue	\fink@nextbase 577	
..... 27	\fink@nextdir 579	P
\@svnmulti@filehooksfalse	\fink@nexttext 578	\pgf@declareimage .
..... 21	\fink@prepare 567, 569, 570	.. 1020, 1022, 1023
\@svnmulti@filehookstrue	\fink@restore 584, 586, 587	\pgf@filename
..... 32	\finkbase 590	.. 1025, 1026, 1030
\@svnmulti@graphicsfalse	\finkdir 592	\pgf@image 1028
..... 17	\finkext 591	\pgf@image@ 1021
\@svnmulti@graphicstrue	\fink@maindir 559	S
..... 28	\fink@mainext 558	\svn 1176
\@svnmulti@groupsfalse		\svn@addfiletogroup
..... 15	 723,
\@svnmulti@groupstrue	G	741, 743, 745,
..... 26	\Gin@setfile 1008, 1010, 1011	746, 913, 971, 972
\@svnmulti@pgfimagesfalse		\svn@catcodes
..... 18	I	. 300, 309, 318,
\@svnmulti@pgfimagestrue	\if@svnidsw ... 243, 350	344, 434, 450, 958
..... 29	\if@svnmulti@anygraphic	\svn@cg 183, 232, 701,
\@svnmulti@subgroupsfalse 10, 992	702, 707, 708, 717
..... 22	\if@svnmulti@autoload	\svn@checkcolon 364, 408
\@svnmulti@subgroupstrue	... 11, 74, 94, 980	\svn@checkwrite ...
..... 33	\if@svnmulti@external 932, 949,
\@svnmulti@tablefalse 57, 919	1200, 1207, 1215
..... 20	\if@svnmulti@filehooks	\svn@cleanfilelist .
\@svnmulti@tabletrue 31 532	785, 855, 1233, 1257
\@svnmulti@verbatimtrue	\if@svnmulti@graphics	\svn@definegroup .. 817
..... 14, 25 64, 103, 1006	\svn@depooption
\@tempa 209,	\if@svnmulti@groups	. 47, 55, 58, 61,
212, 214, 711, 715	.. 54, 276, 572,	62, 65, 66, 67,
\@tempb 210	677, 920, 1213, 1232	70, 71, 72, 75,
\@tempcnta 1285	\if@svnmulti@pgfimages	76, 79, 80, 89, 90
\@temptokena .. 600, 602 69, 106, 1018	\svn@externalfile .
\{ 301, 314, 322	\if@svnmulti@subgroups 964, 966
\} 302, 315, 322 60, 280,	\svn@externalgroup .
\^ . 312, 961, 1152, 1164	872, 902, 911, 968 923, 924,
_ 959, 1150, 1162	\if@svnmulti@table .	926, 928, 937, 998
	... 78, 1037, 1276	\svn@file 1258,
_ 313	\if@svnmulti@verbatim	1259, 1261,
 299, 308,	1262, 1263,
	320, 360, 433, 449	1264, 1266, 1267
E	\IfFileExists 155	\svn@filebase
\endsvnfilerow ... 1143	\ifsvnsubgroups 281, 282, 561,
\endsvnglobalrow . 1137	870, 903, 912, 969	564, 575, 577,
\endsvngrouprow .. 1139		590, 741, 743,
\endsvnsubgrouprow 1141	M	745, 746, 877,
\endsvntable	\message 49,	945, 957, 971, 984
.. 1129, 1301, 1304	567, 584, 1008, 1020	

\svn@filedir 282, 563,	840, 841, 842,	\svn@removedotslash
565, 575, 579,	843, 844, 845, 847 549,
580, 592, 593,	\svn@glst 678, 690,	560, 580, 593, 876
741, 743, 745,	691, 693, 1238,	\svn@removedotslash@
746, 876, 877,	1241, 1243, 1290 550, 552, 556
945, 957, 971, 984	\svn@gobblezeros ..	\svn@restorefilekws
\svn@fileext 562, 564, 202, 203, 205 627, 664
578, 591, 740,	\svn@gobblezeros@ .	\svn@restorefilekws@
742, 744, 904, 983 206, 208 628, 636
\svn@filename . 564, 565	\svn@graphicsgroup .	\svn@restorefilekws@@
\svn@filepath 995, 646, 649
.... 565, 573,	996, 998, 1013, 1030	\svn@rg ... 154, 156, 970
737, 746, 878, 941	\svn@grouplevel 1253,	\svn@s 1176, 1178
\svn@files ... 1255,	1262, 1264,	\svn@scanId ... 245, 353
1256, 1257, 1258	1266, 1285, 1286	\svn@stringa .. 124, 126
\svn@filestack 605,	\svn@ifempty	\svn@stringb .. 125, 126
608, 624, 633,	. 109, 382, 409,	\svn@stripcolon 411, 413
634, 659, 662, 664	414, 632, 936, 1195	\svn@strip space 418, 421
\svn@finkerror	\svn@ifequal 122, 416,	\svn@strip trailing space
.... 535, 539,	422, 551, 573, 424, 426
541, 542, 543,	662, 680, 700,	\svn@subgroup
544, 545, 546, 547	737, 740, 742,	. 877, 878, 881, 973
\svn@g 154,	744, 922, 934,	\svn@svt 1040,
156, 183, 277,	970, 983, 994, 1263	1048, 1062, 1278
278, 290, 574,	\svn@ifvalidrev ...	\svn@svtwrite . 1070,
622, 654, 684,	136, 782, 860, 1259	1277, 1278,
685, 686, 688,	\svn@initfile	1280, 1302, 1307
689, 698, 701, 186, 201, 739	\svn@tabcell .. 1074,
924, 926, 935,	\svn@initgroup	1075, 1092,
995, 1290, 1291, 689, 798, 832	1101, 1110, 1119
1294, 1295, 1296	\svn@input	\svn@tabcellarg 1077,
\svn@gdefverb	152, 161, 1048, 1062	1078, 1079,
.. 429, 1229, 1230	\svn@inputsvx	1080, 1081,
\svn@gdefverb@ 159, 945, 984, 989	1082, 1083,
.... 436, 438, 452	\svn@maindir	1084, 1091,
\svn@getfilename ..	. 559, 560, 563, 737	1093, 1100,
..... 258, 1189	\svn@mainext 558, 562,	1102, 1109,
\svn@gkwdef ... 818,	573, 737, 740, 913	1111, 1118, 1120
819, 820, 821,	\svn@n 1176, 1177	\svn@tabfile 1167, 1169
822, 823, 824,	\svn@namegdefverb .	\svn@tabsubgroup ..
825, 826, 827,	. 443, 768, 773, 775 1155, 1157
828, 829, 830, 851	\svn@nextchar . 206, 211	\svn@temp 86,
\svn@gkwset 802, 803,	\svn@pg ... 183, 574,	87, 92, 93, 111,
804, 805, 806,	575, 623, 655,	112, 139, 141,
807, 808, 809,	685, 741, 743, 745	142, 181, 431,
810, 811, 812,	\svn@popfilestack .	440, 446, 499,
813, 814, 833, 162, 658, 673	500, 503, 504,
834, 835, 836,	\svn@pushfilestack .	520, 521, 524,
837, 838, 839, 160, 607, 669	525, 719, 721,
		781, 782, 785,

786, 787, 789,	\svncgsecond 218	\svnfiletimezone . . 203
859, 861, 862,	\svncgtime 218	\svnfiletimezonehour
890, 891, 895,	\svncgtimezone 204 196,
896, 935, 937,	\svncgtimezonehour . 204	203, 270, 463,
938, 941, 1191, 1192	\svncgtimezoneminute	478, 618, 650, 811
\svn@tmplist 204	\svnfiletimezoneminute
.... 857, 862, 865	\svncgtoday 231 197,
\svn@updategroup . .	\svncgyear 234	203, 271, 464,
. 278, 282, 796, 885	\svnconsiderextensions	479, 619, 651, 812
\svn@value . . . 411, 894	\svnfiletoday 225
415, 417, 423, 427	\svnconsidergraphic	\svnfileurl . . . 198,
\svn@write . . . 750, 1002	257, 258, 272,
766, 771, 940,	\svndate 169, 1219	486, 620, 652, 813
950, 1202, 1203,	\svnday	\svnfileyear . . 190,
1204, 1205,	173, 221, 482, 1223	227, 264, 457,
1216, 1234,	\svnexternal	469, 612, 640, 805
1240, 1247, 1249	. . 930, 1013, 1030	\svnfname . . . 180, 1230
\svn@writeallgroups	\svnexternalfile . .	\svnFullAuthor 493, 1171
778, 788, 1237, 1243 957, 978	\svnFullAuthor@ . .
\svn@writefilerow .	\svnexternalgroup . 921 500, 504, 506
. . 1107, 1262, 1264	\svnexternalpath . . 947	\svnFullAuthor@normal
\svn@writeglobalrow	\svnfileauthor 496, 502
..... 1116, 1287 189, 256,	\svnFullAuthor@star
\svn@writegroup 748, 786	263, 611, 639, 804 495, 498
\svn@writegroupfiles	\svnfiledate 188, 255,	\svnFullRevision . . 514
..... 1251,	262, 610, 638, 803	\svnFullRevision@ .
1267, 1288, 1296	\svnfileday . . . 192, 521, 525, 527
\svn@writegrouprow .	227, 266, 459,	\svnFullRevision@normal
..... 1089, 1295	471, 614, 642, 807 517, 523
\svn@writekw 749, 754,	\svnfilefname	\svnFullRevision@star
755, 756, 757,	. 199, 273, 487, 516, 519
758, 759, 760,	621, 653, 814, 1196	\svng 718
761, 762, 763, 764	\svnfilehour . . 193,	\svng@files 1235
\svn@writerow	217, 267, 460,	\svnglobalrow . . . 1136
... 1069, 1095,	472, 615, 643, 808	\svngraphicsgroup . 993
1104, 1113, 1122	\svnfileminute 194,	\svngroup 679, 682
\svn@writesubgrouprow	217, 268, 461,	\svngrouprow 1138
..... 1098, 1266	473, 616, 644, 809	\svnhour
\svn@writesvn 1212, 1314	\svnfilemonth . 191,	174, 216, 483, 1224
\svn@writesvt 1275, 1315	227, 265, 458,	\svnid 239
\svnafertable	470, 613, 641, 806	\svnidlong 293
..... 1135, 1305	\svnfilerev	\svnidlong@readargs
\svnauthor . . 170, 1220	. 187, 254, 259, 304, 325, 329
\svnbeforetable . . .	260, 261, 609,	\svnidlong@readargsfull
..... 1134, 1281	637, 801, 802, 882 304, 306, 324
\svncg@def . . . 704, 715	\svnfilerow 1142	\svnignoreextensions
\svncgday 234	\svnfilesecond 195, 889, 899
\svncghour 218	217, 269, 462,	\svnignoregraphic . 999
\svncgminute 218	474, 617, 645, 810	\svnk 338, 339, 403
\svncgmonth 234	\svnfiletime 217	\svnkAuthor 395

\svnkwdDate 396	\svnkwsave@read	...	\svnrevisionsname	.
\svnkwddef 330, 331,		... 1039, 1046,	
. 248, 249, 250,		332, 333, 348, 363		1047, 1060, 1061	
251, 294, 295,		\svnkwsave@readargs		\svnsecond
296, 297, 373,	 345, 347		176, 216, 483, 1226	
398, 399, 400,		\svnkURL 397	\svnsetcg 699
401, 402, 410, 411		\svnmainfilename	..	\svnsetmainfile	485, 489
\svnkwddef@ 179, 487		\svnsubgroup
. 375, 378, 383,		\svnmainurl	... 179, 486 874, 905, 914	
384, 386, 387, 388		\svnminute	\svnsubgrouprow	.. 1140
\svnkwddef@Author	..	175, 216, 483, 1225		\svnsubgroupsfalse	. 871
..... 386, 391		\svnmonth	\svnsubgroupstrue	. 873
\svnkwddef@Date	387, 392	172, 221, 482, 1222		\svntabauthor 1171
\svnkwddef@LastChangedAt		\svnmulti@atbegininputfile		\svntabdate	. 1076, 1172
..... 392		. 595, 668, 736, 982		\svntabfile 1158
\svnkwddef@LastChangedBy		\svnmulti@atendinputfile		\svntabglobal 1144
..... 391	 598, 672, 901		\svntabgroup 1145
\svnkwddef@LastChangedRevision		\svnmulti@begininputfilehook		\svntable
..... 390	 566, 581, 596		.. 1125, 1279, 1282	
\svnkwddef@Rev	\svnmulti@endinputfilehook		\svntablefoot	1133, 1303
.... 381, 389, 390		. 583, 588, 600, 602		\svntablehead	1130, 1283
\svnkwddef@Revision	. 389	\svnmulti@fink@prepare		\svntabrev 1170
\svnkwddef@URL 388 569, 571		\svntabs subgroup	.. 1146
\svnkWHeadURL 397	\svnmulti@fink@restore		\svntime 216
\svnkWId 351, 353 586, 589		\svntimezone 202
\svnkWLastChangedAt	396	\svnmulti@Gin@setfile		\svntimezonehour	..
\svnkWLastChangedBy	395 1010, 1015		177, 202, 483, 1227	
\svnkWLastChangedDate		\svnmulti@pgf@declareimage		\svntimezoneminute	.
..... 335, 336	 1022, 1024		178, 202, 483, 1228	
\svnkWLastChangedRevision		\svnnolinkurl	1182, 1185	\svntoday 219
..... 394		\svnnolinkurl@	\svnurl 180, 1229
\svnkWRev 393, 394 1186, 1188		\svnyear
\svnkWRevision 393	\svnpdfdate 481	171, 221, 482, 1221	
\svnkWsave	... 241, 342	\svnRegisterAuthor	. 490		
\svnkWsave@parse	.. 369	\svnRegisterRevision		T	
	 511		\thesvncg 717
		\svnrev 168, 1218	\thesvngroup 698