

The svn-multi Package

Martin Scharrer

martin@scharrer-online.de

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

Version 2.4c

April 1, 2011

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 of the most recent changed file of the document to the user through a set of macros. This information is written to an auxiliary `.aux` 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.

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.

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

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 \LaTeX 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 \TeX groups (`{ }`, `\begingroup \endgroup`), e.g. “in the current group” or “group local”, they will be called *keyword groups*.

There is no limitation (besides internal \LaTeX 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 \TeX 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 \TeX 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 except of the experimental ones.
- 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` with the `french` option.
- external** Controls the support for keywords from external files described in section 2.10. This needs either the external script `svn-multi.pl` or the **autokw** option.

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*.
- 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/graphicx` 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 \LaTeX 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 \LaTeX 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` 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

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 `filehook` 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. A `\clearpage` should be added at the very end of `\included` files to ensure the last page is flushed out by `TeX` before the keyword macros are restored. Otherwise the last page might display the mainfile keyword values.
- autokw** This experimental feature allows the automatically extraction of the keyword values from the hidden Subversion working copy database. The database (a text file called `'entries'`) is located in the hidden Subversion directory `'_svn'` (or `'_svn'` on some Windows installations) inside every directory which is under VC.
- This feature makes an external script like `svn-multi.pl` or even keyword macros redundant as long the files are inside a Subversion working directory. However, this feature does not work if the files are exported (e.g. with `svn export`) or manually copied and also depends on the used version of Subversion. Only versions starting with 1.4 (with working copy format version 7) are supported. Earlier version used a different format for the `entries` file. Newer versions should be compatible as long the basic format is not changed again.
- The experimental status will be lifted after the feature was tested using different Subversion versions on different platforms. Please do not hesitate to send error reports to the package author. Minimal examples and information about the used Subversion version and platform are very much appreciated.
- This option allows the following values:
- false** Feature is disabled (default).
- No value or **true** or **all** The keywords of all files are automatically extracted. No `\svnid` or `\svnidlong` macros or external scripts are necessary as long the files are inside a Subversion working directory and not exported.
- ext** Only the keywords of external files are extracted. This avoids the need of the `svn-multi.pl` script. The option `external` must be still enabled manually.

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: svn-multi.dtx 2431 2011-04-01 23:00:26Z martin $}
```

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: svn+ssh://scharrer-online.de/home/martin/svn/src/trunk/latex/svn-multi/dev/svn-multi.dtx $}  
{$LastChangedDate: 2011-04-02 00:00:26 +0100 (Sat,02 Apr 2011) $}  
{$LastChangedRevision: 2431 $}  
{$LastChangedBy: martin $}
```

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 the revision of the `\svnIdlong` 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.x version load the `currfile` package which lets you typeset the current file name anyway using `\currfilename`². Before v2.3 the `fink` package was used to provide the file names.

This macro must be written like seen above while the order of arguments is not meaningful. 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 `\svnIdlong` 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.

²The file name in `Id` is always the original Subversion file name while the one given by the `currfile` package is the current file name. Both could differ if the file got renamed.

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.

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 \LaTeX 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` 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.

```
\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 \LaTeX 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 \LaTeX 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 `currfile` package. Please see section 2.5 how to typeset parts of the date. See `\svnkW` 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 \LaTeX 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 type-

set with `\svnkw{HeadURL}` or `\svnkw{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 an 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`.

`\svnkwd{<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 `\svnkwd` 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 `\svnkwd{Rev}{123}` and then typeset it with `\svnkw{Revision}` or `\svnkw{LastChangedRev}` if you like.

New in version 2.2. Will change in future versions:

`\ifsvnfilemodified{<case: modified>}{<case: not modified>}` `\ifsvnmodified{<case: modified>}{<case: not modified>}`

This two macro can be used to check if either the current or any file was modified after it was last checked into the repository. At the moment (v2.2) a file is marked 'modified' if there is either a '*' or 'M' after the revision number, e.g. '\$Rev: 123M \$'. Such an

marker is automatically added for exported files (`svn export`) if there where locally modified, but can also be added manually.

Future versions of this package might mark files as modified by use of the `autokw` option which can read this information out of the working directory Subversion entries.

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}`, `\svnidlone` or `\svnidl`, 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. `\svnidl` and `\svnidlone`, 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 \TeX 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 author³ 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}
```

³This means subversion authors, e.g. the persons who commit changes into the svn repository.

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.

```
\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.

	<code>\chapterORsection*{\svnrevisionsname}</code>	
	<code>\svnbeforetable</code> <code>\svntable (like \begin{svntable})</code>	
	<hr/> <code>\svntablehead</code> <hr/>	
<code>\svnglobalrow</code>	<code>\svntabglobal</code>	<code>\svntabdate</code>
<code>\svngrouprow</code>	<code>\svntabgroup {<group>}</code>	<code>{<year>}{<month>}{<day>}</code>
<code>\svnsubgrouprow</code>	<code>\svntabsubgroup {<level>}{<name>}</code>	<code>{<hour>}{<minute>}{<second>}</code>
<code>\svnfilerow</code>	<code>\svntabfile {<level>}{<name>}</code>	<code>{<TZ hour>}{<TZ minute>}</code>
	<hr/> <code>\svntablefoot</code> <hr/>	
	<code>\endsvntable (like \end{svntable})</code> <code>\svnaftertable</code>	
<code>\endsvnglobalrow</code>		<code>\endsvnglobalrow</code>
<code>\endsvngrouprow</code>		<code>\endsvngrouprow</code>
<code>\endsvnsubgrouprow</code>		<code>\endsvnsubgrouprow</code>
<code>\endsvnfilerow</code>		<code>\endsvnfilerow</code>

Figure 1: Table formatting macros and their position.

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.

`\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}{l1111} \end{tabular}` group by default. They can be redefined using

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

In order to support special table packages like `tabularx` or `longtable` 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”.

`\svntablefoot`

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

<code>\svnglobalrow</code>	<code>\endsvnglobalrow</code>
<code>\svngrouprow</code>	<code>\endsvngrouprow</code>
<code>\svnsubgrouprow</code>	<code>\endsvnsubgrouprow</code>
<code>\svnfilerow</code>	<code>\endsvnfilerow</code>

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` 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.

`\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>] {<file 1>}{<file 2>}...{<file n>}`

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 `.aux` 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{<path 1>}{<path 2>}...{<path n>}`

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, `.aux` 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 <jobname>
```

As already mentioned in the `\svnexternal` description above this script reads the requested external filename from the `<jobname>.aux` file. The Subversion command line client `svn` is then used to fetch the needed keywords which are placed in a `\svnidlong` macro inside a `<source file>.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 <jobname> [--group <group name>] <file(s)> ...
```

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 <group name>` 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 `<jobname>.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 <sub file> <external files for sub file>`, which will create/overwrite the `<sub file>.svx` file. However the files given by `\svnexternal` in this sub-file will not be honoured in this case.

```
svn-multi.pl <jobname> [--group <group name>] --fls
```

Instead of providing a list of all non- \LaTeX /external files the `--fls` option can be used to read this list from the `<jobname>.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 `.aux` file is generated. Document global and group keyword macros are not valid yet.

2. Compile document again with \LaTeX . The `.aux` 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 `.aux` 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`. This files contain the keywords of the external files.
3. Compile document again with \LaTeX . The `.svx` files are read by `svn-multi` and the `.aux` file is updated to take the new keywords into account. Document global and group keywords macros only hold internal values.
4. Compile document again with \LaTeX . The `.aux` 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 causes by this package.

4.1 Packet listings uses `\input`

Update: Newer versions of `svn-multi` avoid this issue by changing the catcodes back to normal while reading the `.svx` file. If a file `\langle basenname \rangle.\langle extension \rangle` is typeset verbatim using `\lstinputlisting`, which uses `\input` to read the file, an existing `\langle basenname \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 \LaTeX 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/>.

Since v2.3 the authors packages `currfile` and `filehook` replacing the previous used (and patched) package `fink`.

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>

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, \LaTeX 3 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 \TeX 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: \TeX 3 Project

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

currfile

The file name and path information are taken from the macros of this package. It is from the same author and with the same license as `svn-multi` itself.

Version used: 2011/01/03 v0.3

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

filehook

This package is used to install file hooks to update subversion macros for subfiles etc. It is from the same author and with the same license as `svn-multi` itself.

Version used: 2011/01/03 v0.4

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

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 \PracTeX Journal, (3), 2007. URL: <http://www.tug.org/pracjourn/2007-3/scharrer/>
- [2] Mark Eli Kalderon, “LaTeX and Subversion”, The \PracTeX Journal, (3), 2007. URL: <http://www.tug.org/pracjourn/2007-3/kalderon-svnmulti/>
- [3] Uwe Ziegenhagen, “LaTeX Document Management with Subversion”, The \PracTeX 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 \ProvidesPackageSVN
3   {$Id: svn-multi.dtx 2431 2011-04-01 23:00:26Z ✓
   martin $}
4   [\svnmulti@version\space SVN Keywords for multi-✓
   file LaTeX documents]
```

Options

Declaration of options and internal switches.

```
5 \RequirePackage{kvoptions}
6
7 \SetupKeyvalOptions{%
8     family = svn-multi,
9     prefix = @svnmulti@
10 }
11 \newif\if@svnmulti@anygraphic
12 \newif\if@svnmulti@autoload
13 \newif\if@svnmulti@autokw
14 \newif\if@svnmulti@autokwall
15
16 \DeclareVoidOption{old}{%
17     \@svnmulti@verbatimtrue
18     \@svnmulti@groupsfalse
19     \@svnmulti@externalfalse
20     \@svnmulti@graphicsfalse
21     \@svnmulti@pgfimagesfalse
22     \@svnmulti@autoloadfalse
23     \@svnmulti@tablefalse
24     \@svnmulti@filehooksfalse
25     \@svnmulti@subgroupsfalse
26 }
27 \DeclareVoidOption{all}{%
28     \@svnmulti@verbatimtrue
29     \@svnmulti@groupstrue
30     \@svnmulti@externaltrue
31     \@svnmulti@graphicstrue
32     \@svnmulti@pgfimagestrue
33     \@svnmulti@autoloadtrue
34     \@svnmulti@tabletrue
35     \@svnmulti@filehookstrue
36     \@svnmulti@subgroupstrue
37 }
38 \DeclareBoolOption[true]{verbatim}
39 \DeclareBoolOption[false]{groups}
40 \DeclareBoolOption[false]{external}
41 \DeclareBoolOption[false]{subgroups}
42 \DeclareBoolOption[false]{graphics}
43 \DeclareBoolOption[false]{pgfimages}
44 \DeclareStringOption{autoload}[true]
45 \DeclareBoolOption[false]{table}
46 \DeclareBoolOption[false]{filehooks}
47 \DeclareStringOption[false]{autokw}[all]
48
49 \ExecuteOptions{old}
50 \ProcessKeyvalOptions{svn-multi}
```

Enable dependent options:

```
51 \def\svn@depooption#1{%
52   \csname if@svnmulti@#1\endcsname\else
53   \message{svn-multi: Required option '#1' enabled.}%
54   \csname @svnmulti@#1true\endcsname
55   \fi
56 }
57
58 \if@svnmulti@groups
59   \svn@depooption{filehooks}
60 \fi
61 \if@svnmulti@external
62   \svn@depooption{filehooks}
63 \fi
64 \if@svnmulti@subgroups
65   \svn@depooption{groups}
66   \svn@depooption{filehooks}
67 \fi
68 \if@svnmulti@graphics
69   \svn@depooption{external}
70   \svn@depooption{autoload}
71   \svn@depooption{filehooks}
72 \fi
73 \if@svnmulti@pgfimages
74   \svn@depooption{external}
75   \svn@depooption{autoload}
76   \svn@depooption{filehooks}
77 \fi
78 \if@svnmulti@autoload
79   \svn@depooption{external}
80   \svn@depooption{filehooks}
81 \fi
82 \if@svnmulti@table
83   \svn@depooption{groups}
84   \svn@depooption{filehooks}
85 \fi

```

Check if **autoload** was set explicitly and obey the value.

```
86 \ifx\@svnmulti@autoload\@undefined
87 \else
88 \ifx\@svnmulti@autoload\empty
89 \else
90 \def\svn@temp{true}
91 \ifx\@svnmulti@autoload\svn@temp
92   \@svnmulti@autoloadtrue
93   \svn@depooption{external}
94   \svn@depooption{filehooks}
95 \else
96 \def\svn@temp{false}
```

```

97 \ifx\@svnmulti@autoload\svn@temp
98 \if@svnmulti@autoload
99 \PackageWarning{svn-multi}{Option 'autoload' ✓
    disabled.}
100 \fi
101 \@svnmulti@autoloadfalse
102 \else
103 \PackageError{svn-multi}%
104 {Invalid value for 'autoload' option: '\✓
    @svnmulti@autoload'^~J%
105 ! Only 'true', 'false' or empty ('true') are ✓
    allowed!}{%}
106 \fi\fi\fi\fi

    Set autokw modes:

107 \def\svn@temp{true}
108 \ifx\@svnmulti@autokw\svn@temp
109 \@svnmulti@autokwtrue
110 \@svnmulti@autokwalltrue
111 \svn@depooption{filehooks}
112 \fi
113 \def\svn@temp{all}
114 \ifx\@svnmulti@autokw\svn@temp
115 \@svnmulti@autokwtrue
116 \@svnmulti@autokwalltrue
117 \svn@depooption{filehooks}
118 \fi
119 \def\svn@temp{ext}
120 \ifx\@svnmulti@autokw\svn@temp
121 \@svnmulti@autokwtrue
122 \@svnmulti@autokwallfalse
123 \fi
124 \def\svn@temp{false}
125 \ifx\@svnmulti@autokw\svn@temp
126 \@svnmulti@autokwfalse
127 \@svnmulti@autokwallfalse
128 \fi

    General switch if any graphic option is enabled:

129 \if@svnmulti@graphics
130 \@svnmulti@anygraphictrue
131 \fi
132 \if@svnmulti@pgfimages
133 \@svnmulti@anygraphictrue
134 \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.

```
135 \def\svn@ifempty#1{%
136   \begingroup
137   \edef\svn@temp{#1}%
138   \ifx\svn@temp\empty
139     \endgroup
140     \expandafter
141     \@firstoftwo
142   \else
143     \endgroup
144     \expandafter
145     \@secondoftwo
146   \fi
147 }
```

`\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.

```
148 \def\svn@ifequal#1#2{%
149   \begingroup
150   \edef\svn@stringa{#1}%
151   \edef\svn@stringb{#2}%
152   \ifx\svn@stringa\svn@stringb
153     \endgroup
154     \expandafter
155     \@firstoftwo
156   \else
157     \endgroup
158     \expandafter
159     \@secondoftwo
160   \fi
161 }
```

`\svn@ifvalidrev`

#1: macro name

Checks if the given macro (by name) is a valid revision, i.e. defined and not equal to the init value.

```

162 \def\svn@ifvalidrev#1{%
163   \begingroup
164   \@ifundefined{#1}%
165     {\let\svn@temp\svn@revinit}%
166     {\expandafter\edef
167       \expandafter\svn@temp\expandafter{\csname #1\
168         \ifnum\svn@temp=\svn@revinit\relax
169         \endgroup
170         \expandafter
171         \@secondoftwo
172       \else
173         \endgroup
174         \expandafter
175         \@firstoftwo
176       \fi
177   }

```

\svn@ifeof

#1: input file handle
 Checks if the input file is at the end-of-file (or not open).

```

178 \def\svn@ifeof#1{%
179   \ifeof#1%
180     \expandafter\@firstoftwo
181   \else
182     \expandafter\@secondoftwo
183   \fi
184 }

```

\svn@ifonlyone

#1: group name
 Checks if there is only one element in the given group file list. It looks whether there is a comma in the list.

```

185 \def\svn@ifonlyone#1{%
186   \expandafter\expandafter\expandafter
187   \svn@@ifonlyone\csname @svng@#1@files\endcsname ,\
188   relax
189 }
190 \def\svn@@ifonlyone#1,#2\relax{%
191   \svn@ifempty{#2}
192 }

```


`\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 filehook package and there is no need to use filehook for this files.

```
193 \def\svn@input#1{%
194   \begingroup
195   \let\svn@rg\svn@g
196   \IfFileExists{#1}{\@@input #1\relax}{}%
197   \global\let\svn@g\svn@rg
198   \endgroup
199 }
```

`\svn@inputsvx`

#1: file name/path without extension

Used to save and restore file keywords when reading .svx files. The normal catcodes are restored to avoid issues in special situations regarding input of verbatim files (e.g. `\lstinputlisting` from the listings package) or other cases where catcodes might have changes (e.g. '%' in .dtx files).

```
200 \def\svn@inputsvx#1{%
201   \svn@pushfilestack
202   \begingroup
203   \svn@normalcatcodes
204   \svn@input{#1.svx}%
205   \endgroup
206   \svn@popfilestack
207 }
```

`\svn@normalcatcodes`

Sets the default catcodes.

```
208 \def\svn@normalcatcodes{%
209   \catcode'\=0\relax
210   \catcode'\{=1\relax
211   \catcode'\}=2\relax
212   \catcode'\$=3\relax
213   \catcode'\&=4\relax
214   \catcode'\^M=5\relax
215   \catcode'\#=6\relax
216   \catcode'\^=7\relax
217   \catcode'\_ =8\relax
218   \catcode'\ =10\relax
219   \catcode'\@=12\relax
```

```

220 \catcode \~ =13\relax
221 \catcode \% =14\relax
222 }

```

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 -2, 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 \TeX run. They are initialised here and defined in the `.aux` 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 `\svnid` and `\svnidlong` macros. They change during the document and are used to produce the values of the `\svn<kw>` macros when the `.aux` file is written.

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

```

223 % Init values
224 \def\svn@revinit{-2}
225 \let\svnrev\svn@revinit \let\@svn@rev\svn@revinit
226 \let\ifsvnmodified\@secondoftwo
227 \def\@svn@modified{@secondoftwo}%
228 \def\svndate{} \def\@svn@date{}
229 \def\svnauthor{} \def\@svn@author{}
230 \def\svnyear{0000} \def\@svn@year{0000}
231 \def\svnmonth{00} \def\@svn@month{00}
232 \def\svnday{00} \def\@svn@day{00}
233 \def\svnhour{00} \def\@svn@hour{00}
234 \def\svnminute{00} \def\@svn@minute{00}
235 \def\svnsecond{00} \def\@svn@second{00}
236 \def\svntimezonehour{+00} \def\@svn@timezonehour{
  +00}
237 \def\svntimezoneminute{00} \def\@svn@timezoneminute{
  00}
238 \def\svnmainurl{NOT SET} \def\svnmainfilename{NOT ✓
  SET}
239 \def\svnurl{} \def\svnfname{}
240 \def\svn@temp{}
241
242 \def\svn@pg{} \def\svn@g{} \def\svncg{\svn@g} \def\✓
  svn@rg{\svn@pg}
243 \let\@svng@@files\empty
244
245 \def\svn@initfile{%

```

```

246 \global\let\svnfilerev\svn@revinit
247 \global\let\ifsvnfilemodified\@secondoftwo
248 \gdef\svnfiledate{}%
249 \gdef\svnfileauthor{}%
250 \gdef\svnfileyear{0000}%
251 \gdef\svnfilemonth{00}%
252 \gdef\svnfileday{00}%
253 \gdef\svnfilehour{00}%
254 \gdef\svnfileminute{00}%
255 \gdef\svnfilesecond{00}%
256 \gdef\svnfiletimezonehour{+00}%
257 \gdef\svnfiletimezoneminute{00}%
258 \gdef\svnfileurl{}%
259 \gdef\svnfilename{}%
260 \gdef\svnfiledir{}%
261 }
262 \svn@initfile
263
264 \newif\ifsvn@modified

```

7.4 Auto-Keywords

Special care must be taken for the line feed character, otherwise it causes an error if `autokw` is disabled.

```

265 \begingroup
266 \@makeother\^^L
267 \if@svnmulti@autokw
268 \gdef\svne@ff{\^^L}
269 \fi
270 \endgroup
271
272 \if@svnmulti@autokw
273
274 \newread\svne@read

```

\svne@catcodes

Sets the catcodes for verbatim input reading. Also removes the end-of-line character.

```

273 \newcommand*\svne@catcodes{%
274 \let\do\@makeother
275 \endlinechar=-1%
276 \dospecials
277 \do0\do1\do2\do3\do4%
278 \do5\do6\do7\do8\do9%
279 \do\:\do\^^L%
280 \do\<\do\>\do*\do\.\do\-%
281 \do\/\do\[\do\]\do\'\do\'%
282 \def\do##1{\catcode'\##1=11\relax}%

```

```

283 \do\A\do\B\do\C\do\D\do\E\do\F
284 \do\G\do\H\do\I\do\J\do\K\do\L
285 \do\M\do\N\do\O\do\P\do\Q\do\R
286 \do\S\do\T\do\U\do\V\do\X\do\Y\do\Z
287 \do\a\do\b\do\c\do\d\do\e\do\f
288 \do\g\do|h\do|i\do|j\do|k\do|l
289 \do|m\do\n\do\o\do\p\do\q\do\r
290 \do\s\do\t\do\u\do\v\do\x\do\y\do\z
291 }

```

\svne@readline

#1: macro

Reads the next line to the provided macro and handles the end-of-file case correctly.

```

292 \def\svne@readline#1{%
293   \ifeof\svne@read
294   \def#1{}%
295   \else
296   \read\svne@read to #1\relax
297   \fi
298 }

```

\svne@gobblertest

Gobbles the rest of the current entry.

```

299 \def\svne@gobblertest{%
300   \ifeof\svne@read
301   \let\next\relax
302   \else
303   \read\svne@read to \svn@temp
304   \ifx\svn@temp\svne@ff
305   \let\next\relax
306   \else
307   \let\next\svne@gobblertest
308   \fi
309   \fi
310   \next
311 }

```

\svne@endread

Stops the reading process of the entries file.

```

312 \def\svne@endread{%
313   \closein\svne@read
314 }

```

`\svne@parseentriesfile`

#1: file path

```
315 \newcommand*{\svne@parseentriesfile}[1]{%
316   \begingroup
317   \let\next\relax
```

Open the format file to read the version number. If this file does not exist (true for recent svn versions) a valid default value is used and the true version number is read from the entries file.

```
318   \def\svne@version{8}%
319   \openin\svne@read=#1format\relax
320   \ifeof\svne@read\else
321     \svne@readline\svne@version
322   \closein\svne@read
323   \fi
```

Check the format version:

```
324   \ifnum\svne@version>7\relax
```

Now open the entries file and read the version number from there again.

```
325     \openin\svne@read=#1entries\relax
326     \ifeof\svne@read\else
327       \svne@readline\svne@version
```

Check the version and call the parse macros if OK:

```
328     \ifnum\svne@version>7\relax
329       \def\next{\svne@parsedirentury
330         \svne@parseentries}%
331     \else
332       \closein\svne@read
333     \fi
334   \fi
335 \fi
336 \next
337 \endgroup
338 }
```

`\svne@parsedirentury`

Reads the first entry which is the directory entry and sets its URL as base URL for all other entries.

```
339 \newcommand*{\svne@parsedirentury}{%
340   \svne@readline\svne@name
341   \svne@readline\svne@kind
342   \svn@ifempty{\svne@name}%
343   {\svn@ifequal{\svne@kind}{dir}}%
```

```

344     {%
345         {\svne@readline\svn@temp}%
346         \svne@readline\svne@baseurl
347         \svne@gobblertest
348     }{}%
349 }{}%
350 }

```

\svne@scandate

\svne@scandate@

Parses the date from the svn entries file. Special care is taken to handle the case when the TeX parsing would fail. The catcode of the characters '-', ':', '.' used inside the date is set explicitly to ensure the correct value.

```

351 \begingroup
352
353 \@makeother\~
354 \@makeother\:
355 \@makeother\.
356
357 \gdef\svne@scandate#1{%
358     \expandafter\svne@scandate@#1\empty
359     0000-00-00T00:00:00.00000Z\empty\empty
360 }
361
362 \gdef\svne@scandate@#1-#2-#3T#4:#5:#6.#7\empty#8\
363     empty{%
364     \xdef\svnfileyear{#1}%
365     \gdef\svnfilemonth{#2}%
366     \gdef\svnfileday{#3}%
367     \gdef\svnfilehour{#4}%
368     \gdef\svnfileminute{#5}%
369     \gdef\svnfilesecond{#6}%
370     \gdef\svnfiletimezonehour{+00}%
371     \gdef\svnfiletimezoneminute{00}%
372     \gdef\svnfiledate{#1-#2-#3 #4:#5:#6Z}%
373     \def\svne@date{#1-#2-#3 #4:#5:#6Z}%
374 }
375 \endgroup

```

\svne@parseentries

```

376 \newcommand*{\svne@parseentries}{%
377   \svn@ifeof{\svne@read}%
378   {}%
379   {%
380     \svne@readline\svne@name
381     \@onelevel@sanitize\svne@name
382     \svn@ifeof{\svne@read}%
383     {}%
384     {%
385       \svne@readline\svne@kind
386       \svn@ifequal{\svne@kind}{file}%
387       {%
388         \svne@readline\svn@temp
389         \svne@readline\svn@temp
390         \svne@readline\svn@temp
391         \svne@readline\svn@temp
392         \svne@readline\svn@temp
393         \svne@readline\svn@temp
394         \svne@readline\svne@date
395         \svne@readline\svne@rev
396         \svne@readline\svne@author
397         %\@onelevel@sanitize\svne@date
398         \svne@scandate{\svne@date}%
399         \edef\svne@url{\svne@baseurl/\svne@name}%
400         \svne@handleentry
401         }{}%
402         \svne@gobblerest
403         \svne@parseentries
404       }%
405     }%
406 }

```

\svne@handleentry

This macro is called for every entry except the first one which stands for the directory. The VC data is located in the following macros: \svne@name, \svne@date, \svne@rev, \svne@author, \svne@url.

This implementation sets the correct keywords and calls the update macro to emulate the behaviour of \svnidlong. Then the \svne@endread macro is used to stop the file reading.

```

407 \def\svne@handleentry{%
408   \ifx\svne@rev\empty
409     \let\svne@rev\svn@revinit
410   \fi
411   \svn@ifequal{\svne@name}{\svnfilename}%
412   {%
413     \message{^^J%
414       Read from '.svn/entries' file:^^J%

```

```

415     Filename:  \svne@name^^J%
416     Date:      \svne@date^^J%
417     Revision:  \svne@rev^^J%
418     Author:    \svne@author^^J%
419     HeadURL:   \svne@url^^J%
420     ^^J%
421   }%
422   \svnkundef{Filename}{\svne@name}%
423   \svnkundef{Date}{\svne@date}%
424   \svnkundef{Revision}{\svne@rev}%
425   \svnkundef{Author}{\svne@author}%
426   \svnkundef{HeadURL}{\svne@url}%
427   \@svn@updateid{\svne@rev}{\svne@date}{\svne@author}{\svne@url}%
428   \svne@endread
429 }{}%
430 }%

```

<code>\svnegetfile</code>

#1: file path

```

431 \def\svnegetfile#1{%
432   \begingroup
433     \svn@getfilename{#1}%
434     \edef\svnfilename{\svnfilename}%
435     \@onelevel@sanitize\svnfilename
436     \svne@catcodes
437     \svne@parseentriesfile{\svnfiledir .svn/}%
438     \svne@parseentriesfile{\svnfiledir _svn/}%
439   \endgroup
440 }

```

Load keywords of main document at begin of the document body if option is set to 'all'.

```

441 \if@svnmulti@autokwall
442 \AtBeginDocument{%
443   \svnegetfile{\jobname.\currfile@mainext}%
444 }
445 \fi
446 \fi

```

7.5 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.

```
447 \def\svntimezone{\svntimezonehour\svntimezoneminute\svn@gobblezeros}
448 \def\svnfiletimezone{\svnfiletimezonehour\svnfiletimezoneminute\svn@gobblezeros}
449 \def\svncgtimezone{\svncgtimezonehour\svncgtimezoneminute}
```

`\svn@gobblezeros`

`\svn@gobblezeros@`

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

```
450 \def\svn@gobblezeros{%
451   \futurelet\svn@nextchar\svn@gobblezeros@
452 }
453 \def\svn@gobblezeros@{%
454   \let\@tempa=\relax
455   \def\@tempb{0}%
456   \ifx0\svn@nextchar
457     \let\@tempa=\@gobbletwo
458   \fi
459   \@tempa
460 }
```

`\svntime`

`\svnfiletime`

`\svncgtime`

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

```
461 \def\svntime{\svnhour:\svnminute:\svnsecond}
462 \def\svnfiletime{\svnfilehour:\svnfileminute:\sv
    svnfilesecond}
463 \def\svncgtime{\svncghour:\svncgminute:\svncgsecond}
```

7.6 *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`

```
464 \newcommand*\svntoday}{%
465   \begingroup
466     \year\svnyear \month\svnmonth \day\svnday
467     \relax \today
468   \endgroup
469 }
```

`\svnfiletoday`

```
470 \newcommand*\svnfiletoday}{%
471   \begingroup
472     \year\svnfileyear \month\svnfilemonth \day\sv
    svnfileday
473     \relax \today
474   \endgroup
475 }
```

`\svncgtoday`

```
476 \newcommand*\svncgtoday}{%
477   \@ifundefined{svng@\svn@c @year}{??}{%
478     \begingroup
479       \year\svncgyear \month\svncgmonth \day\svncgday
480       \relax \today
481     \endgroup
482   }%
483 }
```

7.7 Id macros

7.7.1 Normal Id

`\svnid`

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

```
484 \newcommand*{\svnid}{%
485   \@svnidswtrue
486   \svnkwsave
487 }
488 \newif\if@svnidsw
489 \@svnidswfalse
```

`\svn@scanId`

#1: file name
#2: rest of id line

Ensures proper behaviour also for copied but not yet committed files which have no date, time and author name. The revision is set to -1. So `<rest>` is simply compared to '-1' and a fall-back text is supplied if needed.

```
490 \def\svn@scanId#1 #2\relax{%
491   \begingroup
492   \def\@tempa{#2}%
493   \def\@tempb{-1}%
494   \ifx\@tempa\@tempb
495     \endgroup
496     \svn@scanId@#1 -1 0000-00-00 00:00:00Z (✓
         uncommitted)\relax
497   \else
498     \endgroup
499     \svn@scanId@#1 #2\relax
500   \fi
501 }
```

`\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.

```

502 \def\svn@scanId@#1 #2 #3 #4 #5\relax{%
503   \@svn@scandate{#3 #4}%
504   \svnkwdef{Filename}{#1}%
505   \svnkwdef{Date}{#3 #4}%
506   \svnkwdef{Revision}{#2}%
507   \svnkwdef{Author}{#5}%
508   \@svn@updateid{\svnkw{Revision}}{\svnkw{Date}}{\svnkw{Author}}{\svnkw{URL}}%
509 }

```

\@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.

```

510 \def\@svn@updateid#1#2#3#4{%
511   \begingroup
512   \let\protect\@unexpandable@protect
513   \xdef\svnfilerev{#1}%
514   \ifsvn@modified
515     \global\let\ifsvnfilemodified\@firstoftwo
516   \else
517     \global\let\ifsvnfilemodified\@secondoftwo
518   \fi
519   \xdef\svnfiledate{#2}%
520   \xdef\svnfileauthor{#3}%
521   \xdef\svnfileurl{#4}%
522   \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 .aux-file later.

```

523   \ifx\svnfilerev\empty\else
524     \ifnum\@svn@rev<\svnfilerev
525       \xdef\@svn@rev{\svnfilerev}%
526       \xdef\@svn@modified{\ifsvnfilemodified{\@firstoftwo}{@secondoftwo}}%
527       \xdef\@svn@date{\svnfiledate}%
528       \xdef\@svn@author{\svnfileauthor}%
529       \xdef\@svn@year{\svnfileyear}%
530       \xdef\@svn@month{\svnfilemonth}%
531       \xdef\@svn@day{\svnfileday}%
532       \xdef\@svn@hour{\svnfilehour}%
533       \xdef\@svn@minute{\svnfileminute}%

```

```

534     \xdef\@svn@second{\svnfilesecond}%
535     \xdef\@svn@timezonehour{\svnfiletimezonehour}%
536     \xdef\@svn@timezoneminute{\
svnfiletimezoneminute}%
537     \xdef\@svn@url{\svnfileurl}%
538     \xdef\@svn@fname{\svnfilefname}%
539 \fi
540
541 \if@svnmulti@groups
542   \ifx\svn@g\empty\else
543     \svn@updategroup{\svn@g}%
544   \fi
545   \if@svnmulti@subgroups
546     \ifsvnsubgroups
547       \svn@updategroup{\currfiledir\currfilebase}
%
548     \fi
549   \fi
550 \fi
551 \fi
552 \endgroup
553 }
554
555 \def\@svncg@save#1#2{%
556   \expandafter\xdef\csname @svng@\svn@g @#1\endcsname
{#2}%
557 }

```

7.7.2 Long Id

`\svnidlong`

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

```

559 \newcommand{\svnidlong}{%
560   \svnkundef{URL}{}%
561   \svnkundef{Date}{}%
562   \svnkundef{Revision}{0}%
563   \svnkundef{Author}{}%
%
% Read arguments verbatim or non-verbatim.
564   \if@svnmulti@verbatim
565     \expandafter\svnidlong@readverb
566   \else
567     \expandafter\svnidlong@readargs
568   \fi
569 }

```

`\svnidl@readverb`

The following macros read the four arguments of `\svnidl` one-by-one with verbatim mode deactivated between them to ignore all comments. The macro `\@ifnextchar` is used to get rid of all spaces (and therefore comments) between the arguments. An error message is printed if a wrong syntax is discovered.

```
570 \def\svnidl@readverb{%  
571   \@ifnextchar\bgroup  
572     {\svnidl@readverb@\svnidl@readverb@a}%  
573     {\PackageError{svn-multi}{Wrong syntax for \  
       string\svnidl}{}}%  
574 }
```

Sets up verbatim mode and calls the macro given as an argument.

```
575 \def\svnidl@readverb@a#1{%  
576   \begingroup  
577   \svn@catcodes  
578   \catcode`\{=1\relax  
579   \catcode`\}=2\relax  
580   #1%  
581 }
```

Reads first argument, ignores spaces and comments and calls next macro.

```
582 \def\svnidl@readverb@a#1{%  
583   \endgroup  
584   \svnkwsave@read #1\relax  
585   \@ifnextchar\bgroup  
586     {\svnidl@readverb@\svnidl@readverb@b}%  
587     {\PackageError{svn-multi}{Wrong syntax for \  
       string\svnidl}{}}%  
588 }
```

Reads second argument, ignores spaces and comments and calls next macro.

```
589 \def\svnidl@readverb@b#1{%  
590   \endgroup  
591   \svnkwsave@read #1\relax  
592   \@ifnextchar\bgroup  
593     {\svnidl@readverb@\svnidl@readverb@c}%  
594     {\PackageError{svn-multi}{Wrong syntax for \  
       string\svnidl}{}}%  
595 }
```

Reads third argument, ignores spaces and comments and calls next macro.

```
596 \def\svnidl@readverb@c#1{%  
597   \endgroup  
598   \svnkwsave@read #1\relax  
599   \@ifnextchar\bgroup  
600     {\svnidl@readverb@\svnidl@readverb@d}%
```

```

601     {\PackageError{svn-multi}{Wrong syntax for \✓
        string\svnidlong}{}}%
602 }

```

Reads last argument, scans date if not empty and calls the Id update macro.

```

603 \def\svnidlong@readverb@d#1{%
604     \endgroup
605     \svnkwsave@read #1\relax
606     \ifx\svnkwdate\empty\else
607         \@svn@scanlongdate{\svnkwdate}%
608     \fi
609     \@svn@updateid{\svnkwd{Revision}}{\svnkwd{Date}}%
610     {\svnkwd{Author}}{\svnkwd{URL}}%
611     \ignorespaces
612 }

```

\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.

```

613 \if@svnmulti@verbatim
614 \def\svn@catcodes{%
615     \let\do\@makeother
616     \dospecials
617     \catcode'\^^M9
618     \catcode'\ 10
619     \catcode'\{1
620     \catcode'\}2
621 }
622 \else
623 \def\svn@catcodes{}
624 \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.

```

625 \def\svnidlong@readargs#1#2#3#4{%
626     \svnkwsave@read #1\relax
627     \svnkwsave@read #2\relax
628     \svnkwsave@read #3\relax
629     \svnkwsave@read #4\relax
630     \endgroup

```

Now the update macros for date and id are called.

```
631 \ifx\svnkwDate\empty\else
632   \@svn@scanlongdate{\svnkwDate}%
633 \fi
634 \@svn@updateid{\svnkw{Revision}}{\svnkw{Date}}%
635 {\svnkw{Author}}{\svnkw{URL}}%
636 \ignorespaces
637 }%
```

7.8 KeyWord Macros

`\svnkwsave`

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

```
638 \def\svnkwsave{%
639   \begingroup
640   \svn@catcodes
641   \svnkwsave@readargs
642 }
```

`\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.

```
643 \gdef\svnkwsave@readargs#1{%
644   \svnkwsave@read#1\relax
645 \endgroup
646 \if@svnidsw
647   \ifx\svnkwId\empty\else
648     \expandafter
649     \svn@scanId\svnkwId\relax
650     \@svnidswfalse
651   \fi
652 \fi
653 \ignorespaces
654 }
```

`\svnkwsave@read`

#1: keyword line without surrounding \$ \$

Reads the full keyword and strips the dollars.


```

655 \begingroup
656 \if@svnmulti@verbatim
657 \catcode '\$=12
658 \fi
659 \gdef\svnkwsave@read $#1$\relax{%
660   \svn@checkcolon#1:\relax
661 }
662 \endgroup

```

\svnkwsave@parse

#1: key
#2: value
Parse the keyword and save it away.

```

663 \begingroup
664 \catcode '\$=11
665 \gdef\svnkwsave@parse $#1:#2${%
666   \expandafter\xdef\csname svnk#1\endcsname {#2}%
667 }%
668 \endgroup

```

\svnkwdef

#1: key
#2: value
First we check if there is a ‘setter’-macro for the keyword called `\svnkwdef@<keyword>`.

```

669 \newcommand{\svnkwdef}[2]{%
670   \@ifundefined{svnkwdef@#1}%
671     {\svnkwdef@{#1}{#2}}%
672     {\csname svnkwdef@#1\endcsname {#2}}%
673 }

```

\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.

```

674 \newcommand{\svnkundef@}[2]{%
675   \begingroup
676   \let\protect\@unexpandable@protect
677   \expandafter\xdef\csname svnk#1\endcsname{#2}%
678   \endgroup
679 }

```

Example: `\svnkundef{Revision}{23}` will define `\svnkRevision` as 23.

`\svnkundef@Rev`

`\svnkundef@Author`

`\svnkundef@Date`

`\svnkundef@URL`

#1: value

'Setter'-macros for single keywords, used by `\svnkundef`.

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 `\svnkundef@` with a fixed first argument. For the revision the value is checked if empty and then a 0 is substituted. Also a temp counter is used to strip any trailing characters like 'M' which indicate an exported and modified file.

```

680 \def\svnkundef@Rev#1{%
681   \svn@ifempty{#1}%
682     {\svnkundef@{Rev}{0}}%
683     {%
684       \afterassignment\svnkundef@Rev@
685       \@tempcnta=#1\relax
686     }%
687 }
688 \def\svnkundef@Rev@#1\relax{%
689   \svnkundef@{Rev}{\the\@tempcnta}%
690   \def\svn@temp{#1}%
691   \if M\svn@temp\relax
692     \global\svn@modifiedtrue
693   \else
694     \if *\svn@temp\relax
695       \global\svn@modifiedtrue
696     \else
697       \global\svn@modifiedfalse
698     \fi
699   \fi

```

```

700 }
701 \def\svnkwdef@Author#1{\svnkwdef@{Author}{#1}}
702 \def\svnkwdef@Date#1{\svnkwdef@{Date}{#1}}
703 \def\svnkwdef@URL#1{\svnkwdef@{HeadURL}{#1}}

```

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

```

704 \let\svnkwdef@Revision=\svnkwdef@Rev
705 \let\svnkwdef@LastChangedRevision=\svnkwdef@Rev
706 \let\svnkwdef@LastChangedBy=\svnkwdef@Author
707 \let\svnkwdef@LastChangedDate=\svnkwdef@Date

```

and then for reading.

```

708 \def\svnkwRevision{\svnkwRev}
709 \def\svnkwLastChangedRevision{\svnkwRev}
710 \def\svnkwLastChangedBy{\svnkwAuthor}
711 \def\svnkwLastChangedDate{\svnkwDate}
712 \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.

```

713 \svnkwdef{Rev}{0}
714 \svnkwdef{Date}{}
715 \svnkwdef{Author}{}
716 \svnkwdef{Filename}{}
717 \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.

```

718 \newcommand{\svnkw}[1]{%
719   \@ifundefined{svnkw#1}%
720     {\PackageWarning{svn-multi}{SVN keyword '#1' not
       defined (typo?)}}%
721     {\csname svnkw#1\endcsname}%
722 }%

```

7.9 Keyword check and strip macros

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

macros. Unexpanded keywords result in an empty value. Also leading and trailing spaces are removed.

`\svn@checkcolon`

#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`.

```
723 \def\svn@checkcolon#1:#2\relax{%  
724   \svn@ifempty{#2}%  
725   {\svnkwdef{#1}{}}%  
726   {\svn@stripcolon#2\relax\svnkwdef{#1}{\svn@value  
    }}%  
727 }
```

`\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@stripspaces` is called to strip the spaces.

```
728 \def\svn@stripcolon#1:\relax{%  
729   \svn@ifempty{#1}%  
730   {\gdef\svn@value{}}%  
731   {\svn@ifequal{#1}{ }%  
732   {\gdef\svn@value{}}%  
733   {\svn@stripspaces#1\relax\relax}%  
734   }%  
735 }
```

`\svn@stripspaces`

#1: first character

#2: rest of string

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

```
736 \def\svn@stripspaces#1#2\relax{%  
737   \svn@ifequal{#1}{ }%  
738   {\gdef\svn@value{#2}}%  
739   {\svn@striptrailingspace#1#2\relax}%  
740 }
```

`\svn@striptrailingspace`

#1: string

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

```
741 \def\svn@striptrailingspace#1 \relax{%  
742   \gdef\svn@value{#1}%  
743 }
```

`\svn@gdefverb`

#1: macro

```
744 \def\svn@gdefverb#1{%  
745   \begingroup  
746     \def\svn@temp{#1}%  
747     \begingroup  
748       \if@svnmulti@verbatim  
749         \svn@catcodes  
750       \fi  
751     \svn@gdefverb@  
752 }
```

`\svn@defverb@`

#1: verbatim stuff

```
753 \def\svn@gdefverb@#1{%  
754   \endgroup  
755   \expandafter\gdef\svn@temp{#1}%  
756   \endgroup  
757 }
```

`\svn@namegdefverb`

#1: macro name

```
758 \def\svn@namegdefverb#1{%  
759   \begingroup  
760     \expandafter\def  
761     \expandafter\svn@temp  
762     \expandafter{\csname #1\endcsname}%  
763     \begingroup  
764       \if@svnmulti@verbatim  
765         \svn@catcodes  
766       \fi  
767     \svn@gdefverb@  
768 }
```

7.10 Date Macros

`\@svn@scandate`

#1: date

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

```
769 \def\@svn@scandate#1{\@svn@scandate@#1\relax}
770
771 \def\@svn@scandate@#1-#2-#3 #4:#5:#6#7#8\relax{%
772   \gdef\svnfileyear{#1}%
773   \gdef\svnfilemonth{#2}%
774   \gdef\svnfileday{#3}%
775   \gdef\svnfilehour{#4}%
776   \gdef\svnfileminute{#5}%
777   \gdef\svnfilesecond{#6#7}%
778   \gdef\svnfiletimezonehour{+00}%
779   \gdef\svnfiletimezoneminute{00}% #8 always 'Z' for ✓
      Zulu-time (UTC)
780 }
```

`\@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.

```
781 \def\@svn@scanlongdate#1{\expandafter\✓
      \@svn@scanlongdate@#1\relax}
782 %
783 \def\@svn@scanlongdate@#1-#2-#3 #4:#5:#6 #7 #8\relax{✓
      %
784   \gdef\svnfileyear{#1}%
785   \gdef\svnfilemonth{#2}%
786   \gdef\svnfileday{#3}%
787   \gdef\svnfilehour{#4}%
788   \gdef\svnfileminute{#5}%
789   \gdef\svnfilesecond{#6}%
790   \@svn@parsetimezone#7\relax%
791 }
```

`\@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.

```
792 \def \@svn@parsetimezone #1#2#3#4#5 \relax {%  
793   \gdef \svnfiletimezonehour {#1#2#3}%  
794   \gdef \svnfiletimezoneminute {#4#5}%  
795 }
```

`\svnpdfdate`

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

```
796 \def \svnpdfdate {%  
797   \svnyear \svnmonth \svnday  
798   \svnhour \svnminute \svnsecond \svntimezonehour ' \sv  
       svntimezoneminute '%  
799 }
```

7.11 Mainfile Makros

`\svnsetmainfile`

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

```
800 \newcommand {\svnsetmainfile} {%  
801   \xdef \svnmainurl {\svnfileurl}%  
802   \xdef \svnmainfilename {\svnfilename}%  
803 }  
804 \AtBeginDocument {\svnsetmainfile}
```

7.12 Register and FullName Macros

`\svnRegisterAuthor`

#1: author username

#2: Full Name

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

```

805 \newcommand{\svnRegisterAuthor}[2]{%
806   \expandafter\def\csname svn@author@#1\endcsname{#2}✓
      %
807 }

```

`\svnFullAuthor`

`\svnFullAuthor*`

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

```

808 \newcommand{\svnFullAuthor}{%
809   \@ifnextchar{*}%
810     {\svnFullAuthor@star}%
811     {\svnFullAuthor@normal}%
812 }%

```

`\svnFullAuthor@star`

#1: username

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

```

813 \def\svnFullAuthor@star*#1{%
814   \edef\svn@temp{#1}%
815   \svnFullAuthor@{\svn@temp}{~(\svn@temp)}%
816 }%

```

`\svnFullAuthor@normal`

#1: username

```

817 \def\svnFullAuthor@normal#1{%
818   \edef\svn@temp{#1}%
819   \svnFullAuthor@{\svn@temp}{}%
820 }%

```

`\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.


```

821 \def\svnFullAuthor@#1#2{%
822   \@ifundefined{svn@author@#1}%
823     {#1}%
824     {\csname svn@author@#1\endcsname #2}%
825 }

```

\svnRegisterRevision

#1: revision number
 #2: tag name

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

```

826 \newcommand{\svnRegisterRevision}[2]{%
827   \expandafter\def\csname svn@revision@#1\endcsname
828     {#2}%
829 }

```

\svnFullRevision

\svnFullRevision*

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

```

829 \newcommand{\svnFullRevision}{%
830   \@ifnextchar{*}%
831     {\svnFullRevision@star}%
832     {\svnFullRevision@normal}%
833 }

```

\svnFullRevision@star

#1: revision number

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

```

834 \def\svnFullRevision@star*#1{%
835   \edef\svn@temp{#1}%
836   \svnFullRevision@{\svn@temp}{~(r\svn@temp)}%
837 }

```

\svnFullRevision@normal

#1: revision number

```

838 \def\svnFullRevision@normal#1{%
839   \edef\svn@temp{#1}%
840   \svnFullRevision@{\svn@temp}{}%
841 }

```

<code>\svnFullRevision@</code>

#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.

```

842 \def\svnFullRevision@#1#2{%
843   \@ifundefined{svn@revision@#1}%
844     {Revision #1}%
845     {\csname svn@revision@#1\endcsname #2}%
846 }

```

7.13 Input File Name

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

```

847 \if@svnmulti@filehooks
      Load filehook and currfile packages.
848 \RequirePackage{filehook}[2011/01/03]
849 \RequirePackage{currfile}[2011/01/03]

```

The following code installs the necessary hooks for subfiles. It installs an own file stack currfile does. The macro \svn@pg for the parent group needs to be set before \currfilepath etc. is updated, so the internal filehook macros are used to install the code before the currfile code. This means that the file name macros below still hold the values of the parent file.

```

850 \filehook@prefixwarg\filehook@every@atbegin{%
851   \svn@pushfilestack
852   \if@svnmulti@groups
853     \svn@ifequal{\currfilepath}{\jobname.\✓
      currfile@mainext}%
854     {\xdef\svn@pg{\svn@g}}%
855     {\xdef\svn@pg{\currfiledir\currfilebase}}%
856   \fi
857 }

```

The file stack is popped at the end of the file hook to keep the macros valid for normal hook code.

```

858 \filehook@appendwarg\filehook@every@atend{%
859   \svn@popfilestack
860 }

```

```

861 \def\svn@filestack{ $\{\}$ }
862
863 \def\svn@pushfilestack{ $\%$ 
864   \xdef\svn@filestack{ $\%$ 
865     {\svnfilerev} $\%$ 
866     {\svnfiledate} $\%$ 
867     {\svnfileauthor} $\%$ 
868     {\svnfileyear} $\%$ 
869     {\svnfilemonth} $\%$ 
870     {\svnfileday} $\%$ 
871     {\svnfilehour} $\%$ 
872     {\svnfileminute} $\%$ 
873     {\svnfilesecond} $\%$ 
874     {\svnfiletimezonehour} $\%$ 
875     {\svnfiletimezoneminute} $\%$ 
876     {\svnfileurl} $\%$ 
877     {\svnfilename} $\%$ 
878     {\svn@g} $\%$ 
879     {\svn@pg} $\%$ 
880     {\ifsvnfilemodified{@firstoftwo}{@secondoftwo}} $\%$ 
881   }\svn@filestack} $\%$ 
882 }
883
884 \def\svn@restorefilekws#1#2\relax{ $\%$ 
885   \svn@restorefilekws@#1\empty
886   \empty \empty \empty \empty
887   \empty \empty \empty \empty
888   \empty \empty \empty \empty \empty
889   \svn@ifempty{#2} $\%$ 
890     {\gdef\svn@filestack{ $\{\}$ }} $\%$ 
891     {\gdef\svn@filestack{#2}} $\%$ 
892 }
893 \def\svn@restorefilekws@#1#2#3#4#5#6#7#8#9{ $\%$ 
894   \gdef\svnfilerev{#1} $\%$ 
895   \gdef\svnfiledate{#2} $\%$ 
896   \gdef\svnfileauthor{#3} $\%$ 
897   \gdef\svnfileyear{#4} $\%$ 
898   \gdef\svnfilemonth{#5} $\%$ 
899   \gdef\svnfileday{#6} $\%$ 
900   \gdef\svnfilehour{#7} $\%$ 
901   \gdef\svnfileminute{#8} $\%$ 
902   \gdef\svnfilesecond{#9} $\%$ 
903   \svn@restorefilekws@@
904 }
905
906 \def\svn@restorefilekws@@#1#2#3#4#5#6#7{ $\%$ 
907   \gdef\svnfiletimezonehour{#1} $\%$ 
908   \gdef\svnfiletimezoneminute{#2} $\%$ 
909   \gdef\svnfileurl{#3} $\%$ 
910   \gdef\svnfilename{#4} $\%$ 

```

```

911 \gdef\svn@g{#5}%
912 \gdef\svn@pg{#6}%
913 \expandafter\global\expandafter\let
914 \expandafter\ifsvnfilemodified\csname#7\endcsname%
915 }
916
917 \def\svn@popfilestack{%
918 \ifx\svn@filestack\empty
919 \PackageWarning{svn-multi}{Underflow of file ✓
    keyword stack!}%
920 \else
921 \svn@ifequal{\svn@filestack}{{}}%
922 {\PackageWarning{svn-multi}{Underflow of file ✓
    keyword stack!}}%
923 {\expandafter\svn@restorefilekws\svn@filestack\✓
    relax}%
924 \fi
925 }
926
927 %
928 \fi

```

7.14 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`.

```

929 \if@svnmulti@groups
930 \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.

```

931 \def\svngroup#1{%
932 \svn@ifequal{#1}{*}%
933 {\PackageError{svn-multi}%
934 {The group name '*' is invalid for '\string\✓
    svngroup'}{}}%
935 }{}%
936 \xdef\svn@g{#1}%
937 \let\svn@pg\svn@g

```

```

938   \svn@checkgroup{#1}%
939 }
940 \def\svn@checkgroup#1{%
941   \begingroup
942   \edef\svn@g{#1}%

   Only initialise the group at first usage:
943   \ifx\svn@g\empty\else%
944     \expandafter
945     \ifx\csname @svng@\svn@g @rev\endcsname\relax%
946       \svn@initgroup{\svn@g}%

```

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

```

947   \ifx\svn@glist\empty
948     \xdef\svn@glist{#1}%
949   \else
950     \xdef\svn@glist{\svn@glist ,#1}%
951   \fi
952 \fi
953 \fi
954 \endgroup
955 }

```

`\thesvngroup`

Returns the current group name to the user.

```

956 \def\thesvngroup{\svn@g}

```

`\svnsetcg`

#1: group name

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

```

957 \def\svnsetcg#1{%
958   \svn@ifequal{#1}{*}%
959   {\def\svn@cg{\svn@g}}%
960   {\def\svn@cg{#1}}%
961 }

```

`\svncg@def`

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

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

```
962 \def\svncg@def#1{%  
963   \expandafter  
964   \def\csname svncg#1\endcsname{%  
965     \@ifundefined{svng@\svn@cg @#1}{??}{%  
966     \csname svng@\svn@cg @#1\endcsname}%  
967   }%  
968 }
```

\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.

```
969 \@for\@tempa :=%
970     rev , author , date , year , month , day , hour , minute , second , %
971     timezonehour , timezoneminute , url , fname %
972 \do{%
973     \expandafter\svncg@def\expandafter{\@tempa}%
974 }
```

`\thesvncg`

Simply return the internal macro.

```
975 \def\thesvncg{\svn@cg}
```

`\svng`

#1: group name

#2: keyword name

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

```
976 \def\svng#1#2{%
977     \@ifundefined{svng@\svn@temp @#2}%
978     {??}%
979     {\csname svng@\svn@temp @#2\endcsname}%
980 }
```

`\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.

```
981 \def\svn@addfiletogroup#1#2{%
982     \svn@ifequal{#1}{#2}{%
983         \expandafter
984         \ifx\csname @svng@#2@files@#1\endcsname\relax%
985             \expandafter\gdef\csname @svng@#2@files@#1\
986                 \endcsname{1}%
987         }%
988 }
```

```

987 \expandafter\ifx\csname @svg@#2@files\endcsname\
    empty%
988 \expandafter\xdef\csname @svg@#2@files\
    endcsname{#1}%
989 \else
990 \@ifundefined{@svg@#2@files}%
991 {\expandafter\xdef\csname @svg@#2@files\
    endcsname{#1}}%
992 {\expandafter\xdef\csname @svg@#2@files\
    endcsname{%
993 \csname @svg@#2@files\endcsname ,#1%
994 }%
995 }%
996 \fi
997 \fi
998 }%
999 }

```

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}`).

```

1000 \AtBeginOfFiles{%
1001 \svn@ifequal{\currfilepath}{\jobname.\
    currfile@mainext}%
1002 {}%
1003 {\svn@initfile}%
1004 \svn@ifequal{\currfileext}{\currfile@mainext}%
1005 {\svn@addfiletogroup{\currfiledir\currfilebase}{\
    svn@pg}}}%
1006 \svn@ifequal{\currfileext}{sty}%
1007 {\svn@addfiletogroup{\currfiledir\currfilebase}{\
    svn@pg}}}%
1008 \svn@ifequal{\currfileext}{cls}%
1009 {\svn@addfiletogroup{\currfiledir\currfilebase}{\
    svn@pg}}}%
1010 \svn@addfiletogroup{\currfilepath}{\currfiledir\
    currfilebase}%
1011 }

```

`\svn@writegroup`

#1: group name

Writes group to `\svn@write` file.

```

1012 \def\svn@writegroup#1{%
1013 \def\svn@writekw##1{%
1014 \immediate\write\svn@write{%
1015 \string\global\string\@namedef{svg@#1@##1}{\
    csname @svg@#1@##1\endcsname}%

```



```

1016 }%
1017 }%
1018 \svn@writekw{rev}%
1019 \svn@writekw{date}%
1020 \svn@writekw{author}%
1021 \svn@writekw{year}%
1022 \svn@writekw{month}%
1023 \svn@writekw{day}%
1024 \svn@writekw{hour}%
1025 \svn@writekw{minute}%
1026 \svn@writekw{second}%
1027 \svn@writekw{timezonehour}%
1028 \svn@writekw{timezoneminute}%
1029 \@ifundefined{@svng@#1@files}{}{%
1030 \immediate\write\svn@write{%
1031 \noexpand
1032 \svn@namegdefverb{svng@#1@files}{\csname @svng@#1
#1@files\endcsname}%
1033 }%
1034 }%
1035 \immediate\write\svn@write{%
1036 \noexpand
1037 \svn@namegdefverb{svng@#1@url}{\csname @svng@#1
@url\endcsname}^^J%
1038 \noexpand
1039 \svn@namegdefverb{svng@#1@fname}{\csname @svng@#1
@fname\endcsname}^^J%
1040 }%
1041 }

```

\svn@writeallgroups

#1: macro holding a list of groups

```

1042 \def\svn@writeallgroups#1{%
1043 \begingroup
1044 \ifx\relax#1\relax\else
1045 \@for\svn@temp:=#1\do{%
1046 \svn@ifvalidrev{@svng@\svn@temp @rev}%
1047 {%
1048 \expandafter
1049 \svn@cleanfilelist\csname @svng@\svn@temp
@files\endcsname
1050 \svn@writegroup{\svn@temp}%
1051 \@ifundefined{@svng@\svn@temp @files}{}%
1052 {\expandafter\svn@writeallgroups
1053 \csname @svng@\svn@temp @files\
endcsname
1054 }%

```

```

1055     }{}%
1056     }%
1057     \fi
1058     \endgroup
1059 }

```

\svn@updategroup

#1: group name

Updates group with \svnfile... macro values.

```

1060 \def\svn@updategroup#1{%
1061   \@ifundefined{@svng@#1@rev}%
1062     {\svn@initgroup{#1}}%
1063     {}%
1064   \expandafter
1065   \ifnum\csgname @svng@#1@rev\endcsgname<\svnfilerev
1066     \svn@gkwset{#1}{rev}{\svnfilerev}%
1067     \svn@gkwset{#1}{date}{\svnfiledate}%
1068     \svn@gkwset{#1}{author}{\svnfileauthor}%
1069     \svn@gkwset{#1}{year}{\svnfileyear}%
1070     \svn@gkwset{#1}{month}{\svnfilemonth}%
1071     \svn@gkwset{#1}{day}{\svnfileday}%
1072     \svn@gkwset{#1}{hour}{\svnfilehour}%
1073     \svn@gkwset{#1}{minute}{\svnfileminute}%
1074     \svn@gkwset{#1}{second}{\svnfilesecond}%
1075     \svn@gkwset{#1}{timezonehour}{\
1076       svnfiletimezonehour}%
1076     \svn@gkwset{#1}{timezoneminute}{\
1077       svnfiletimezoneminute}%
1077     \svn@gkwset{#1}{url}{\svnfileurl}%
1078     \svn@gkwset{#1}{fname}{\svnfilefname}%
1079   \fi
1080 }

```

\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.

```

1081 \def\svn@definegroup#1{%
1082   \svn@gkwdef{#1}{rev}%
1083   \svn@gkwdef{#1}{date}%
1084   \svn@gkwdef{#1}{author}%
1085   \svn@gkwdef{#1}{year}%
1086   \svn@gkwdef{#1}{month}%
1087   \svn@gkwdef{#1}{day}%

```

```

1088 \svn@gkwdef{#1}{hour}%
1089 \svn@gkwdef{#1}{minute}%
1090 \svn@gkwdef{#1}{second}%
1091 \svn@gkwdef{#1}{timezonehour}%
1092 \svn@gkwdef{#1}{timezoneminute}%
1093 \svn@gkwdef{#1}{url}%
1094 \svn@gkwdef{#1}{fname}%
1095 }

```

\svn@initgroup

#1: group name
Initialises group.

```

1096 \def\svn@initgroup#1{%
1097 \svn@gkwset{#1}{rev}{\svn@revinit}%
1098 \svn@gkwset{#1}{date}{}%
1099 \svn@gkwset{#1}{author}{}%
1100 \svn@gkwset{#1}{year}{0000}%
1101 \svn@gkwset{#1}{month}{00}%
1102 \svn@gkwset{#1}{day}{00}%
1103 \svn@gkwset{#1}{hour}{00}%
1104 \svn@gkwset{#1}{minute}{00}%
1105 \svn@gkwset{#1}{second}{00}%
1106 \svn@gkwset{#1}{timezonehour}{+00}%
1107 \svn@gkwset{#1}{timezoneminute}{00}%
1108 \svn@gkwset{#1}{url}{}%
1109 \svn@gkwset{#1}{fname}{}%
1110 }

```

\svn@gkwset

#1: group name
#2: keyword name
#3: value
Sets *<value>* for *<keyword>* in *<group>*.

```

1111 \def\svn@gkwset#1#2#3{%
1112 \expandafter
1113 \xdef\csname @svng@#1@#2\endcsname{#3}%
1114 }

```

\svn@gkwdef

#1: group name
#2: keyword name
Defines svng@... macros used by the user macros to the value of the internal @svng@... macros.

```

1115 \def\svn@gkwdef#1#2{%
1116   \expandafter
1117   \xdef\csname svng@#1@#2\endcsname{\csname @svng@#1@
      #2\endcsname}%
1118 }

```

\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.

```

1119 \def\svn@cleanfilelist#1{
1120   \begingroup
1121   \def\svn@tmplist{}%
1122   \ifx\relax#1\relax\else
1123     \@for\svn@temp:=#1\do{%
1124       \expandafter\svn@ifvalidrev
1125       \expandafter{@svng@\svn@temp @rev}%
1126       {\edef\svn@tmplist{\svn@tmplist,\svn@temp}}%
1127       }%
1128     }%
1129     \xdef#1{\expandafter\@gobble\svn@tmplist\empty}%
1130   \fi
1131   \endgroup
1132 }
1133 \fi

```

7.15 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.

```

1134 \newif\ifsvnsubgroups
1135 \svnsubgroupsfalse

1136 \if@svnmulti@subgroups
1137 \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.

```
1138 \def\svnsubgroup{%
1139   \begingroup
1140   \svn@subgroup{\currfiledir\currfilebase}%
1141   \svn@subgroup{\currfilepath}%
1142   \endgroup
1143 }
```

`\svn@subgroup`

#1: file name

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

```
1144 \def\svn@subgroup#1{%
1145   \ifnum\svnfilerev=\svn@revinit\else
1146     \expandafter\ifx\csname svn@g@#1\endcsname\relax%
1147     \expandafter\gdef\csname svn@g@#1\endcsname{1}%
1148     \svn@updategroup{#1}%
1149     \fi
1150   \fi
1151 }
```

`\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.

```
1152 \def\svnignoreextensions#1{%
1153   \@for\svn@temp:=#1\do{%
1154     \expandafter\def\csname svn@ignore@ext@\svn@temp\
1155       endcsname{}%
1156   }%
1157 }
```

`\svnconsiderextensions`

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

The special macro defined by `\svnignoreextentions` is deleted, i.e. `\let` to `\relax`.

```

1157 \def\svnconsiderextensions#1{%
1158   \@for\svn@temp:=#1\do{%
1159     \expandafter\let\csname svn@ignore@ext@\svn@temp\relax
1160     \endcsname\relax%
1161   }%

```

The following extensions are ignored by default.

```

1162 \svnignoreextensions{aux,bbl,fd,enc,fls,glo,idx,ilg,
1163   ind,ist,%
1164   lof,log,lot,out,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.

```

1164 \AtEndOfFiles{%
1165   \if@svnmulti@subgroups
1166     \ifsvnsubgroups
1167       \expandafter\ifx\csname svn@ignore@ext@\currfileext\endcsname\relax
1168       \svnsubgroup
1169       \fi
1170     \fi
1171   \fi
1172 }

```

The main file is added to the main base name (`\jobname`) subgroup here. This subgroup is added as first element to the active group at begin of the document body.

```

1173 \if@svnmulti@subgroups
1174   \ifsvnsubgroups
1175     \svn@addfiletogroup{\jobname .\currfile@mainext}
1176     {\jobname}%
1177   \svnsubgroup
1178   \fi
1179 \AtBeginDocument{%
1180   \if@svnmulti@subgroups
1181     \ifsvnsubgroups
1182     \@ifundefined{@svng@\svn@g @files@\jobname}%
1183     {%
1184       \@namedef{@svng@\svn@g @files@\jobname}{1}%
1185     \@ifundefined{@svng@\svn@g @files}%
1186     {%
1187       \expandafter
1188       \xdef\csname @svng@\svn@g @files\endcsname
1189       {\jobname}%
1190     }%

```

```

1191         \expandafter
1192         \xdef\csname @svng@\svn@g @files\endcsname
1193             {\jobname,\csname @svng@\svn@g @files\
1194                 \endcsname}%
1195             }%
1196         }{}%
1197         \svnsubgroup
1198     \fi
1199 }
1200 \fi

```

7.16 External Files

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

```
1201 \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.

```

1202 \if@svnmulti@groups
1203 \def\svnexternalgroup#1{%
1204     \svn@ifequal{#1}{*}%
1205     {\def\svn@externalgroup{\svn@pg}}%
1206     {\def\svn@externalgroup{#1}}%
1207 }
1208 \def\svn@externalgroup{\svn@pg}
1209 \else
1210 \def\svn@externalgroup{}
1211 \fi

```

`\svnexternal`

#1: group name

#2: list of filenames in { }

```

1212 \if@svnmulti@autokw
1213 \newcommand*\svnexternal [2] []{%
1214     \svn@pushfilestack
1215     \svn@ifequal{#1}{*}%
1216     {\edef\svn@eg{\svn@pg}}%

```

```

1217     {\svn@ifempty{#1}%
1218      {\edef\svn@eg{\svn@externalgroup}}%
1219      {\edef\svn@eg{#1}}%
1220     }%
1221     \svn@checkgroup{\svn@eg}%
1222     \svne@@external#2\relax
1223     \svn@popfilestack
1224 }
1225
1226 \def\svne@@external#1{%
1227   \ifx\relax#1\empty\else
1228     \svnegetfile{#1}%
1229     \begingroup\svn@externalfile{\svn@eg}{#1}%
1230     \expandafter\svne@@external
1231   \fi
1232 }
1233
1234 \else
1235     Writes the current input file path and its argument as arguments of \@svnexternal
1236     into the .aux file.
1237
1238 \newcommand*\svnexternal [2] [] {%
1239   \if@filesw
1240     \begingroup
1241     \svn@ifequal{#1}{*}%
1242     {\def\svn@temp{\svn@pg}}%
1243     {\svn@ifempty{#1}%
1244      {\def\svn@temp{\svn@externalgroup}}%
1245      {\def\svn@temp{#1}}%
1246     }%
1247     \let\protect\@unexpandable@protect
1248     \immediate\write\svn@write{%
1249       \noexpand\@svnexternal[\svn@temp]{\↵
1250         currfilepath}{#2}%
1251     }%
1252     \endgroup
1253   \fi
1254   \svn@inputsvx{\currfiledir\currfilebase}%
1255 }
1256
1257 \fi

```

<code>\svnexternalpath</code>

#1: list of paths in { }

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

```

1252 \def\svnexternalpath#1{%
1253   \if@filesw

```



```

1254     \begingroup
1255     \let\protect\@unexpandable@protect
1256     \immediate\write\svn@write{%
1257         \noexpand\@svnexternalpath{#1}%
1258     }%
1259     \endgroup
1260 \fi
1261 }

```

\@svnexternal

\@svnexternalpath

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

```

1262 \newcommand*\@svnexternal [3] [] {}
1263 \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.

```

1264 \newcommand*\svnexternalfile [1] [\currfiledir\
1265     currfilebase]{%
1266     \begingroup % TODO: maybe use \svn@catcodes
1267     \catcode '\_ =12
1268     \catcode '\& =12
1269     \catcode '\^ =12
1270     \catcode '\$ =12
1271     \catcode '\# =12
1272     \svn@externalfile{#1}%
1273 }

```

\svn@externalfile

#1: group name

#2: file name

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

```

1273 \def\svn@externalfile#1#2{%
1274     \endgroup
1275     \if@svnmulti@subgroups
1276     \ifsvnsubgroups
1277         \svn@ifequal{#1}{\svn@rg}%

```

```

1278         {\svn@addfiletogroup{#2}{\currfiledir\✓
           currfilebase}}}%
1279         {\svn@addfiletogroup{#2}{#1}}}%
1280     \svn@subgroup{#2}%
1281     \fi
1282 \fi
1283 }

```

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

```

1284 \else
1285     \def\svnexternalfile#1{}%
1286 \fi

```

7.17 Auto loading of .svx files

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

```

1287 \if@svnmulti@autoload
1288
1289 \AtBeginOfFiles{%
1290     \svn@ifequal{\currfileext}{tex}%
1291     {\svn@inputsvx{\currfiledir\currfilebase}}%
1292     {}%
1293 }

```

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

```

1294 %%\AtEndOfPackage{%
1295 \AtBeginDocument{%
1296     \svn@inputsvx{\jobname}%
1297 }
1298 \fi

```

7.18 Support for Graphic Packages

7.18.1 Common Code

```

1299 \if@svnmulti@anygraphic

```

<code>\svngraphicsgroup</code>

#1: graphic group name
 Defines the default group of graphics files. The default is empty which means the current group.

```

1300 \def\svngraphicsgroup#1{%
1301   \svn@ifequal{#1}{*}%
1302     {\def\svn@graphicsgroup{\svn@pg}}%
1303     {\def\svn@graphicsgroup{#1}}%
1304 }
1305 \def\svn@graphicsgroup{\svn@externalgroup}

```

\svnignoregraphic

#1: file name/path

Ignores the given graphic file by defining a special macro.

```

1306 \def\svnignoregraphic#1{%
1307   \expandafter\def\csname svn@ignoregraphic@#1\
1308     \endcsname{}%
1309 }

```

\svnconsidergraphic

#1: file name/path

Deletes the special ignore macro to consider the graphic again.

```

1309 \def\svnconsidergraphic#1{%
1310   \expandafter\let\csname svn@ignoregraphic@#1\
1311     \endcsname\relax%
1312 }

```

1312 \fi

7.18.2 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.

```

1313 \if@svnmulti@graphics
1314 \RequirePackage{graphics}[2006/02/20]

```

\Gin@setfile

#1: ??, not used

#2: ??, not used

#3: graphic file name/path

```

1315 \message{Package svn-multi: patching macro '\string\
1316   Gin@setfile' from the
1317   'graphics' package!}%
1318 \let\svnmulti@Gin@setfile\Gin@setfile

```

```

1318 \renewcommand*\Gin@setfile}[3]{%
1319   \expandafter\ifx\csname svn@ignoregraphic@#3\✓
       endcsname\relax%
1320   \svnexternal[\svn@graphicsgroup]{{#3}}%
1321   \fi
1322   \svnmulti@Gin@setfile{#1}{#2}{#3}%
1323 }
1324 \fi

```

7.18.3 Package pgf

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

```

1325 \if@svnmulti@pgfimages
1326 \RequirePackage{pgf}[2008/01/15]

```

`\pgf@declareimage`

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

```

1327 \message{Package svn-multi: patching macro '\string\✓
       pgf@declareimage' and will
1328 patch generated macros '\string\pgf@image@<name>!' ✓
       from the 'pgf' package!}%
1329 \let\svnmulti@pgf@declareimage\pgf@declareimage
1330 \renewcommand*\pgf@declareimage}[3][[]]{%
1331   \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`.

```

1332   \ifx\pgf@filename\empty\else
1333     \expandafter\ifx\csname svn@ignoregraphic@\✓
       pgf@filename\endcsname\relax%
1334     \expandafter\global\expandafter%
1335     \let\csname svnmulti@pgf@image@#2!\endcsname=\✓
       pgf@image%
1336     \expandafter\xdef\csname pgf@image@#2!\✓
       endcsname{%
1337       \noexpand\svnexternal[\noexpand\✓
       svn@graphicsgroup]{{\pgf@filename}}%
1338       \csname svnmulti@pgf@image@#2!\endcsname
1339     }%
1340     \fi
1341   \fi
1342 }

```

```
1343 \fi
```

7.19 Table of Revisions

```
1344 \if@svnmulti@table
1345 \ifx\tableofcontents\relax\else
```

\svnrevisionsname

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

```
1346 \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.

```
1347 \def\svn@svt{svt}
```

\tableofrevisions

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

The external (i.e. non-svn-multi) if-switches are masked using \ifx and \cename to avoid \TeX if-parsing errors when they are not defined.

```
1348 \AtBeginDocument{%
1349 \ifx\chapter\relax
1350 \let\chapter\@undefined
1351 \fi
1352 \ifx\chapter\@undefined
1353
1354 %% Adapted from the \tableofcontents macro, LaTeX ‘
    article’ class [2005/09/16]
1355 \newcommand\tableofrevisions{%
1356 \section*{\svnrevisionsname
1357 \@mkboth{\MakeUppercase\svnrevisionsname}{\
    MakeUppercase\svnrevisionsname}}%
1358 \svn@input{\jobname .\svn@svt}%
1359 }
1360
1361 \else
1362
1363 %% Adapted from the \tableofcontents macro, LaTeX ‘
    book’ class [2005/09/16]
1364 \newcommand\tableofrevisions{%
```

```

1365 \expandafter\ifx
1366 \csname if@twocolumn\expandafter\endcsname
1367 \csname iftrue\endcsname
1368 \@restonecoltrue\onecolumn
1369 \else
1370 \@restonecolfalse
1371 \fi
1372 \chapter*{\svnrevisionsname
1373 \@mkboth{\MakeUppercase\svnrevisionsname}{\
1374 \MakeUppercase\svnrevisionsname}}%
1375 \svn@input{\jobname .\svn@svt}%
1376 \expandafter\ifx
1377 \csname if@restonecol\expandafter\endcsname
1378 \csname iftrue\endcsname
1379 \twocolumn
1380 \fi
1381 }
1382 \fi
1383 }
1384 \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.

```

1385 \def\svn@writerow#1#2#3{%
1386 \immediate\write\svn@svtwrite{%
1387 \expandafter\noexpand\csname svn#1row\endcsname
1388 \expandafter\noexpand\csname svntab#1\endcsname
1389 {#2}{#3}\space
1390 \@ampersamchar\space
1391 \svn@tabcell{rev}\space\@ampersamchar\space
1392 \svn@tabcell{author}\space\@ampersamchar\space
1393 \noexpand\svntabdate%
1394 \svn@tabcellarg{year}%
1395 \svn@tabcellarg{month}%
1396 \svn@tabcellarg{day}%
1397 \svn@tabcellarg{hour}%
1398 \svn@tabcellarg{minute}%
1399 \svn@tabcellarg{second}%
1400 \svn@tabcellarg{timezonehour}%
1401 \svn@tabcellarg{timezoneminute}%
1402 \space\@backslashchar\@backslashchar

```

```

1402     \expandafter\noexpand\csname endsvn#1row\✓
        endcsname
1403     }%
1404 }

```

\svn@writegrouprow

#1: current group

```

1405 \def\svn@writegrouprow#1{%
1406     \begingroup
1407     \def\svn@tabcellarg##1{{\csname @svng@#1@##1\✓
        endcsname}}%
1408     \def\svn@tabcell##1{\expandafter\noexpand\csname ✓
        svntab##1\endcsname%
1409     \svn@tabcellarg{##1}%
1410     }%
1411     \svn@writerow{group}{#1}{}%
1412 \endgroup
1413 }

```

\svn@writesubgrouprow

#1: grouping level

#2: subgroup name

```

1414 \def\svn@writesubgrouprow#1#2{%
1415     \begingroup
1416     \def\svn@tabcellarg##1{{\csname @svng@#2@##1\✓
        endcsname}}%
1417     \def\svn@tabcell##1{\expandafter\noexpand\csname ✓
        svntab##1\endcsname%
1418     \svn@tabcellarg{##1}%
1419     }%
1420     \svn@writerow{subgroup}{#1}{#2}%
1421 \endgroup
1422 }

```

\svn@writefilerow

#1: grouping level

#2: file name

```

1423 \def\svn@writefilerow#1#2{%
1424     \begingroup
1425     \def\svn@tabcellarg##1{{\csname @svng@#2@##1\✓
        endcsname}}%

```

```

1426     \def\svn@tabcell##1{\expandafter\noexpand\csname ✓
          svntab##1\endcsname%
1427     \svn@tabcellarg{##1}%
1428     }%
1429     \svn@writerow{file}{#1}{#2}%
1430 \endgroup
1431 }

```

`\svn@writeglobalrow`

```

1432 \def\svn@writeglobalrow{%
1433   \begingroup
1434   \def\svn@tabcellarg##1{{\csname @svn@##1\endcsname ✓
     }}%
1435   \def\svn@tabcell##1{\expandafter\noexpand\csname ✓
     svntab##1\endcsname%
1436   \svn@tabcellarg{##1}%
1437   }%
1438   \svn@writerow{global}{}{}%
1439   \endgroup
1440 }

```

7.19.1 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`

```

1441 \def\svntable{%
1442   \begin{tabular}{p{0.425\textwidth}r1l}%
1443   \hline
1444 }

```

`\endsvntable`

```

1445 \def\endsvntable{\hline\end{tabular}}

```

`\svntablehead`

```

1446 \def\svntablehead{%
1447   Name & Rev & Last Author & Last Changed At \\\ ✓
     \hline
1448 }

```


`\svntablefoot`

1449 `\def\svntablefoot{}`

`\svnbeforetable`

1450 `\def\svnbeforetable{}`

`\svnaftertable`

1451 `\def\svnaftertable{\clearpage}`

`\svnglobalrow`

1452 `\def\svnglobalrow{}`

`\endsvnglobalrow`

1453 `\def\endsvnglobalrow{}`

`\svngrouprow`

1454 `\def\svngrouprow{}`

`\endsvngrouprow`

1455 `\def\endsvngrouprow{}`

`\svnsubgrouprow`

1456 `\def\svnsubgrouprow{}`

`\endsvnsubgrouprow`

1457 `\def\endsvnsubgrouprow{}`

`\svnsubgrouprow`

1458 `\def\svnfilerow{}`

`\endsvnfilerow`

1459 `\def\endsvnfilerow{}`

`\svntabglobal`

1460 `\def\svntabglobal{Document}`

`\svntabgroup`

1461 `\def\svntabgroup#1{Group '#1'}`

`\svntabfile`

```
1462 \def\svntabsubgroup#1{%
1463   \raggedright
1464   \addtolength{\leftskip}{#1\medskipamount}%
1465   \begingroup
1466   \catcode'\_ =12
1467   \catcode'\& =12
1468   \catcode'\^ =12
1469   \catcode'\$ =12
1470   \catcode'\# =12
1471   \svn@tabsubgroup
1472 }
1473 \def\svn@tabsubgroup#1{\endgroup Subgroup '\texttt{\small #1}'}
```

`\svntabfile`

```
1474 \def\svntabfile#1{%
1475   \raggedright
1476   \addtolength{\leftskip}{#1\medskipamount}%
1477   \begingroup
1478   \catcode'\_ =12
1479   \catcode'\& =12
1480   \catcode'\^ =12
```

```

1481 \catcode '\$=12
1482 \catcode '\#=12
1483 \svn@tabfile
1484 }
1485 \def\svn@tabfile#1{\endgroup File '\texttt{\small
#1}' }

```

`\svntabrev`

```

1486 \def\svntabrev{}

```

`\svntabauthor`

```

1487 \def\svntabauthor#1{\svnFullAuthor{#1}}

```

`\svntabdate`

```

1488 \def\svntabdate#1#2#3#4#5#6#7#8{%
1489 #1-#2-#3 #4:#5:#6% #7#8%
1490 }
1491 \fi

```

7.20 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.

```

1492 \newcommand{\svn}{\@ifnextchar{*}{\svn@s}{\svn@n}}
1493 \def\svn@n#1{\@svn@n#1}
1494 \def\svn@s*#1{\@svn@s#1}
1495 \def\@svn@n$#1${#1}
1496 \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.

```
1497 %% Adapted from the \url macro of the 'hyperref' ✓
      package.
1498 \DeclareRobustCommand*\svnnolinkurl}{%
1499   \@ifundefined{hyper@normalise}%
1500     {\PackageWarning{svn-multi}{Package hyperref is ✓
      needed for \noexpand
1501     \svnnolinkurl.}}%
1502     {\hyper@normalise\svnnolinkurl@}%
1503   }%
1504 \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.

```
1505 \def\svn@getfilename#1{%
1506   \begingroup
1507   \gdef\svnfiledir{}%
1508   \edef\svn@temp{#1}%
1509   \expandafter\@svn@getfilename\svn@temp/{}\relax
1510 }%
```

`\@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.

```
1511 \def\@svn@getfilename#1/#2\relax{%
1512   \svn@ifempty{#2}%
1513   {%
1514     \endgroup
1515     \gdef\svnfilename{#1}%
1516   }%
1517   {%
1518     \xdef\svnfiledir{\svnfiledir#1/}%
1519     \@svn@getfilename#2\relax
```

```

1520     }%
1521 }%

```

7.21 Write to Auxiliary File

`\svn@write`

Simply an alias for the main aux file write handle.

```

1522 \let\svn@write\@mainaux

```

`\svn@writesvn`

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

```

1523 {\catcode'\&=12
1524 \gdef\@ampersamchar{&}
1525 }
1526 \def\svn@writesvn{%

```

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

```

1527     \if@svnmulti@groups
1528     \fi

```

Write document global values:

```

1529     \immediate\write\svn@write{^^J%
1530     \@percentchar\space Global values:^^J%
1531     \noexpand\gdef\noexpand\svnrev{\@svn@rev}^^J%
1532     \noexpand\global\noexpand\let\noexpand\
1533         ifsvnmodified\@backslashchar\@svn@modified^^\
1534         J%
1535     \noexpand\gdef\noexpand\svndate{\@svn@date}^^J%
1536     \noexpand\gdef\noexpand\svnauthor{\@svn@author\
1537         }^^J%
1538     \noexpand\gdef\noexpand\svnyear{\@svn@year}^^J%
1539     \noexpand\gdef\noexpand\svnmonth{\@svn@month}^^\
1540         J%
1541     \noexpand\gdef\noexpand\svnday{\@svn@day}^^J%
1542     \noexpand\gdef\noexpand\svnhour{\@svn@hour}^^J%
1543     \noexpand\gdef\noexpand\svnminute{\@svn@minute\
1544         }^^J%
1545     \noexpand\gdef\noexpand\svnsecond{\@svn@second\
1546         }^^J%
1547     \noexpand\gdef\noexpand\svntimezonehour{\
1548         @svn@timezonehour}^^J%
1549     \noexpand\gdef\noexpand\svntimezoneminute{\
1550         @svn@timezoneminute}^^J%

```

```

1543     \noexpand\svn@gdefverb\noexpand\svnurl{\✓
        @svn@url}^^J%
1544     \noexpand\svn@gdefverb\noexpand\svnfname{\✓
        @svn@fname}^^J%
1545 }%

```

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

```

1546     \if@svnmulti@groups
1547     \svn@cleanfilelist\@svng@@files
1548     \immediate\write\svn@write{%
1549     \noexpand\gdef\noexpand\svng@@files{\✓
        @svng@@files}^^J%
1550     }%
1551     \svn@writeallgroups\@svng@@files

```

Write keyword group values if groups were specified:

```

1552     \ifx\svn@glist\empty\else
1553     \begingroup
1554     \immediate\write\svn@write{^^J%
1555     \@percentchar\space SVN File Groups: \✓
        svn@glist
1556     }%
1557     \svn@writeallgroups\svn@glist
1558     \endgroup
1559     \fi
1560     \else
1561     \immediate\write\svn@write{^^J}%
1562     \fi
1563 }

```

\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.

```

1564 \def\svn@writegroupfiles#1{%
1565     \begingroup
1566     \advance\svn@grouplevel by 1\relax
1567     \expandafter\let
1568     \expandafter\svn@files\csname @svng@#1@files\✓
        endcsname

```

Stop if no files in list.

```

1569     \ifx\svn@files\relax\else
1570     \svn@cleanfilelist\svn@files
1571     \@for\svn@file:=\svn@files\do{%

```

Check if VC data is set and then if it is a (sub)group or not by looking at the file list.

```

1572     \svn@ifvalidrev{@svng@\svn@file @rev}%
1573     {%
1574     \@ifundefined{@svng@\svn@file @files}%
1575     {\svn@writefilerow{\the\svn@grouplevel
      }\svn@file}}%

```

If subgroup only contains out of a TeX file with the same name print it as file and only as subgroup otherwise. If the subgroup file list is empty the subgroup was generated in error and is not printed at all.

```

1576     {\svn@ifonlyone{\svn@file}%
1577     {\svn@writefilerow{\the\
      svn@grouplevel}}%
1578     {\csname @svng@\svn@file @files\
      endcsname}}%
1579     {\svn@ifempty{\csname @svng@\svn@file
      @files\endcsname}%
1580     {}}%
1581     {%
1582     \svn@writesubgrouprow{\the\
      svn@grouplevel}{\svn@file}%
1583     \svn@writegroupfiles{\svn@file}%
1584     }%
1585     }%
1586     }%
1587     }{}%
1588     }%
1589     \fi
1590     \endgroup
1591     }%

```

`\svn@writesvt`

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

```

1592 \def\svn@writesvt{%
      Write table of revisions if enabled.
1593 \if@svnmulti@table
      Open .svt file and write table header:
1594     \newwrite\svn@svtwrite
1595     \immediate\openout\svn@svtwrite=\jobname.\svn@svt\
      \relax
1596     \@onelevel@sanitize\svntable%
1597     \immediate\write\svn@svtwrite{%
1598     \noexpand\svnbeforetable^J%

```

```

1599     \svntable
1600     \noexpand\svntablehead^^J%
1601 }%

Group rows:

1602     \let\svn@grouplevel\@tempcnta
1603     \svn@grouplevel=0\relax
1604     \svn@writeglobalrow{}%
1605     \svn@writegroupfiles{}%
1606 %
1607     \@for\svn@g:=\svn@glist\do{%
1608     \@ifundefined{@svng@\svn@g @rev}{}%
1609     {%
1610     \expandafter
1611     \ifnum\csname @svng@\svn@g @rev\endcsname>0\relax
1612     \svn@writegrouprow{\svn@g}%
1613     \svn@writegroupfiles{\svn@g}%
1614     \fi
1615     }%
1616 }%

Write table footer and close file:

1618     \@onelevel@sanitize\endsvntable%
1619     \immediate\write\svn@svtwrite{%
1620     \noexpand\svntablefoot^^J%
1621     \endsvntable^^J%
1622     \noexpand\svnaftertable
1623     }%
1624     \immediate\closeout\svn@svtwrite%
1625 \fi
1626 }

Load the keywords of every subfile if the autokw option is enabled and the
extension is not on the ignore list.
This code is placed here because it has to come after the at-begin-input-file code.

1627 \if@svnmulti@autokwall
1628
1629 \AtBeginOfFiles{%
1630     \expandafter
1631     \ifx\csname svn@ignore@ext@\currfileext\endcsname\relax
1632     \svnegetfile{\currfilepath}%
1633     \fi
1634 }
1635
1636 \fi

```

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


```

1637 \AtEndDocument{%
1638   \if@filesw
1639     \ifx\@svn@rev\empty\else
1640     \ifnum\@svn@rev<1\else
1641       \begingroup
1642       \let\protect\@unexpandable@protect
1643       \svn@writesvn
1644       \svn@writesvt
1645       \endgroup
1646     \fi
1647   \fi
1648 \fi
1649 }

```

7.22 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`.

```

1650 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
1651 \ProvidesPackage{svnkw}
1652   [2009/03/27 v2.1 Backward compatibility wrapper for ✓
   svn-multi]
1653 \PackageWarning{svnkw}{The package 'svnkw' got ✓
   renamed to 'svn-multi' and is now
1654 only a backward compatibility wrapper which loads '✓
   svn-multi'. Please adjust
1655 your document preamble to use the new name.}
1656 \RequirePackage{svn-multi}

```