

# The **svn-multi** Package

formerly known as **svnk**

Martin Scharrer

`martin@scharrer-online.de`

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

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

Version 2.1

March 27, 2009

## 1 Introduction

This package allows to typeset version control (VC) information provided by Subversion<sup>1</sup> keywords (e.g. `$Id: ... $`) in  $\LaTeX$  documents which can contain of multiple `.tex` files included using `\include` or `\input`. Subversion is a modern version control system designed to replace its predecessor CVS and uses integers as revision numbers.

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

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

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

---

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

## 1.1 Scope of Keywords

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

### Document Global

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

### Local to Current File

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

### Groups

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

These groups could also be called *file groups*, *keyword groups* or, like in programming languages, *namespaces*. In this manual they will be reference as simple *groups* most the time. In places where they could be confused with T<sub>E</sub>X groups (`{ }`, `\begin group \end group`), e.g. “in the current group” or “group local”, they will be called *keyword groups*.

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

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

## 2 Usage

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

### 2.1 Package Options

Since v2.0 this package provides options to enable only a needed features, e.g. to avoid problems with other packages or save T<sub>E</sub>X memory. For backwards compatibility to pre-2.0 package versions all old features are enabled by default and all new features are disabled to save a little of T<sub>E</sub>X 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**<sup>2</sup> 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*.

---

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

## groups

Controls the keyword groups feature described in section 2.3.

## subgroups

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

## graphics

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

The `graphics` package is loaded if this option is active. If this package is needed with some special options it should be loaded by the L<sup>A</sup>T<sub>E</sub>X 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`<sup>5</sup> 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

---

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

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

<sup>5</sup>CTAN: <http://tug.ctan.org/pkg/pgf>

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

### table

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

### filehooks

This option loads the **fink**<sup>6</sup> 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.

### 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.

---

<sup>6</sup>CTAN: <http://tug.ctan.org/pkg/fink>

## 2.2 Including Subversion Keywords

Subversion keywords are included using `\svnid` or `\svnidlong`. These macros should be written very early in each file, i.e. in the preamble of the main document soon after `\documentclass` and `\usepackage{svn-multi}` and as first in *every* subfile before an `\chapter` or similar macro. They do not create any output. See section 2.4 to learn how to typeset the keyword values.

`\svnid{$Id$}`

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

`\svnidlong`

`{$HeadURL$}`

`{$LastChangedDate$}`

`{$LastChangedRevision$}`

`{$LastChangedBy$}`

Macro for a “long `Id`”. Saves similar values like in ‘`Id`’ but from the above four keywords. The usage of `\svnid` or `\svnidlong` is a matter of taste. The second is more readable inside the code and results in a nicer date and a full URL, not only the filename. However, both can also be used together. In this case the `\svnid` macro should be come last. Because its revision is not higher (but identical) than 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 2.0 version load the `fink` package which lets you typeset the current file name anyway using `\finkbase.\finkext`<sup>7</sup>.

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

Please note that the arguments are read verbatim as long the **verbatim** option is not disabled explicitly. Special precaution are taken to allow spaces, newlines and comments direct after the `\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

---

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

the last dollar. This macro alone was the very first version of `svnkw` and is still included for fast and simple keyword typesetting.

`\svnkwsave{${keyword}$}`

This macro lets you include and save any keyword you like. The keyword can be already expanded or not (no value and only “:” or nothing after the key name). This macro is also used internally and does not create any output. Please note that the argument is read verbatim and that there should be no space between the macro and the argument’s left brace.

## 2.3 Groups

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

`\svngroup{<group name>}`

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

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

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

`\thesvngroup`

Returns the name of the current keyword group.

`\svnsetcg{<group name>}`

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

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

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

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

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

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

### `\thesvncg`

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

#### 2.3.1 Files as Subgroups

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

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

### `\svnsubgroup`

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

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

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

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

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

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



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

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

## 2.4 Typesetting the Keyword Values

The following macros can be used to typeset the keyword values anywhere in the document. Please note that not all L<sup>A</sup>T<sub>E</sub>X fonts have all special characters, e.g. ‘\_’ is not provided in the standard roman font. To properly typeset file names and URLs containing these letters you can use either teletype font (`\texttt`) or use `{\urlstyle{rm}\svnnolinkurl{...}}` which requires the `hyperref`<sup>8</sup> package.

Like already mentioned `svn-multi` knows three scopes of keywords. The first contains the keywords for the complete document which hold the values of the most recent committed file and the second contains the *current* or *file local* keywords, e.g. the keywords of the current file. Only these 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 L<sup>A</sup>T<sub>E</sub>X document, after `\usepackage{svn}` of course. Please see section 2.5 how to typeset parts of the date.

`\svnfilerev`

`\svnfiledate`

`\svnfileauthor`

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

---

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

`\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  $\text{\LaTeX}$  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 typeset with `\svnk{HeadURL}` or `\svnk{Filename}` after `\svnid` or `\svnidlong`, respectively).

`\svnsetmainfile`

This will declare the current file as the main  $\text{\LaTeX}$  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.

`\svnk{<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 sub-version keyword name. It then returns the current value of this keyword or nothing (`\relax`) when the keyword was not set yet. Examples:

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

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

In the second example `\url` ([hyperref](#) package) is used to add a hyperlink and to avoid problems with underscores (`_`) inside the URL. `svn-multi` is also providing

a macro `\svnnolinkurl` which works like `\url` but doesn't add 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`.

`\svnkwdef{<keyword name>}{<value>}`

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

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

## 2.5 Accessing Date Values

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

Whenever the date information is read, i.e. by `\svnkwsave{LastChangedDate}` `\svnkwsave{Date}`, `\svnidlong` or `\svnid`, 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. `\svnid` and `\svnidlong`, 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  $\text{\LaTeX}$  document.

`\svntoday`

`\svnfiletoday`

`\svncgtoday`

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

## 2.6 Using Full Author Names

If you like to have the full author<sup>9</sup> 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}`.

---

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

`\svnRegisterAuthor{<author>}{<full name>}`

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

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

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

Takes the username as argument and returns the full name if it was registered first with `\svnRegisterAuthor`, otherwise it returns the given username. The star version returns the username in parentheses after the full name. This is normally used in one of the following forms:

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

## 2.7 Using Full Revision Names

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

`\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{\svnkwl{HeadURL}}` and get the Head URL typeset verbatim. However `\url{\svnkwl{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`.

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

Figure 1: Table formatting macros and their position.

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 LATEX file`.`.svt` (t for table) which can be included using the `\tableofrevisions` macro. The table contains the revision information of the complete document and of all defined groups. If the **subgroups** option is set the table will also include the single files sorted by group.

### Table Format Macros

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

### `\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`<sup>10</sup> or `longtable`<sup>11</sup> 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>\svngroupro</code>	<code>\endsvngroupro</code>
<code>\svnsubgroupro</code>	<code>\endsvnsubgroupro</code>
<code>\svnfilerow</code>	<code>\endsvnfilerow</code>

This 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.

---

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

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

`\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`<sup>12</sup> 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>]{<filea>}{<fileb>}\dots{<filex>}`

Subversion keywords of external files (e.g. non- $\text{\LaTeX}$  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  $\text{\LaTeX}$  file and each enclosed by { }. The files must be under version control by Subversion, of course. Use `\svnexternalpath` to specify paths to be scanned for this files if they are not located relative to the main file. The requested filenames are written into the `.svn` auxiliary file and then processed by the external script `svn-multi.pl` which must be executed like described below. The appropriate keywords are then written in `<source file>.svx` files (x like eXternal) which are read in by the same `\svnexternal` macro at the next  $\text{\LaTeX}$  run. If a group name of ‘\*’ for the external files is specified using the

---

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



optional argument the keywords will be placed in the same keyword group as this macro. The file local macros like `\svnfilerev` which appear in a source file after `\svnexternal` are affected, i.e. updated if one of the external revision is higher than the one of the source file. This makes sense if e.g. the included graphics are taken as logical part of a source file.

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

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

## Script `svn-multi.pl`

The file `svn-multi.pl` which comes with the `svn-multi` package is an external Perl script which has to be run in the command line or by a  $\text{\LaTeX}$  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  $\text{\LaTeX}$  document, `.svn` file is generated.
2. Run `svn-multi.pl` script, `.svx` files are generated.
3. Compile  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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>.svn` file. The Subversion command line client `svn` is then used to fetch the needed keywords which are placed in a `\svnidlong` macro inside a `<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 `.svn` file is generated. Document global and group keyword macros are not valid yet.
2. Compile document again with `LATEX`. The `.svn` file is read by `svn-multi`. Document global and group keyword macros are now valid.

#### Table of Revisions

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

#### External Files

1. Compile document with `LATEX`. The `.svn` file is generated with references to the external files. Document global and group keyword macros are not valid yet. External files are not taken into account.
2. Run `svn-multi.pl` with the main base name (main file name without extension) as argument. This generates `.svx` files for each `.tex` file which used `\svnexternal`. These files contain the keywords of the external files.

3. Compile document again with L<sup>A</sup>T<sub>E</sub>X. The `.svx` files are read by `svn-multi` and the `.svn` file is updated to take the new keywords into account. Document global and group keywords macros only hold internal values.
4. Compile document again with L<sup>A</sup>T<sub>E</sub>X. The `.svn` file is read by `svn-multi`. Document global and group keyword macros are now fully valid.

## 4 Known Issues

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

### 4.1 Packet listings uses `\input`

If a file `<basename>.<extension>` is typeset verbatim using `\lstinputlisting`, which uses `\input` to read the file, an existing `<basename>.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 these packages, their internal macro which got used and acknowledge the authors/maintainers of them. Please send error reports to the author of `svn-multi` and not to the people listed below.

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

### **hyperref**

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

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

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

**Licence:** LPPL, any version

**Authors/Maintainers:** Sebastian Rahtz, Heiko Oberdiek

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

## **fink**

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

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

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

**Version used:**    2008/02/27 v2.1.1

**Licence:**    LPPL, any version

**Author/Maintainer:**    Didier Verna

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

## **graphics**

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

**Patched internal macros:**    \Gin@setfile

**Version used:**    2006/02/20 v1.0o

**Licence:**    LPPL, any version

**Author/Maintainer:**    David Carlisle, L<sup>A</sup>T<sub>E</sub>X3 Project

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

## **pgf**

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

**Patched internal macros:**    \pgf@declareimage, \pgf@image@*<image name>*!

**Used internal macros:**    \pgf@filename, \pgf@image

**Version used:**    2008/01/15 v2.00

**Licence:**    LPPL v1.3c and GPL v2

**Author&Maintainer:**    Till Tantau

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

## **latex**

Parts of the macro definitions of the \tableofcontents macros from the **article** and **book** class of standard L<sup>A</sup>T<sub>E</sub>X were used to define a similar **\tableofrevisions** macro for both this classes and other similar classes.

**Version used:** 2005/09/16 v1.4f  
**Licence:** LPPL v1.3c  
**Authors/Maintainers:** L<sup>A</sup>T<sub>E</sub>X3 Project  
**Location:** CTAN: <http://tug.ctan.org/pkg/latex>

## 6 Further Reading

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

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

## 7 Implementation

### 7.1 Package Header

#### Package Identification

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

#### Options

Declaration of options and internal switches.

```
4 \RequirePackage{kvoptions}
5
6 \SetupKeyvalOptions{%
7   family = svn-multi,
8   prefix = @svnmulti@
9 }
10 \newif\if@svnmulti@anygraphic
11 \newif\if@svnmulti@autoload
12 \newif\if@svnmulti@autokw
13 \newif\if@svnmulti@autokwall
14
15 \DeclareVoidOption{old}{%
16   \@svnmulti@verbatimtrue
17   \@svnmulti@groupsfalse
```

```

18 \@svnmulti@externalfalse
19 \@svnmulti@graphicsfalse
20 \@svnmulti@pgfimagesfalse
21 \@svnmulti@autoloadfalse
22 \@svnmulti@tablefalse
23 \@svnmulti@filehooksfalse
24 \@svnmulti@subgroupsfalse
25 }
26 \DeclareVoidOption{all}{%
27 \@svnmulti@verbatimtrue
28 \@svnmulti@groupstrue
29 \@svnmulti@externaltrue
30 \@svnmulti@graphicstrue
31 \@svnmulti@pgfimagestrue
32 \@svnmulti@autoloadtrue
33 \@svnmulti@tabletrue
34 \@svnmulti@filehookstrue
35 \@svnmulti@subgroupstrue
36 }
37 \DeclareBoolOption[true]{verbatim}
38 \DeclareBoolOption[false]{groups}
39 \DeclareBoolOption[false]{external}
40 \DeclareBoolOption[false]{subgroups}
41 \DeclareBoolOption[false]{graphics}
42 \DeclareBoolOption[false]{pgfimages}
43 \DeclareStringOption[autoload]{true}
44 \DeclareBoolOption[false]{table}
45 \DeclareBoolOption[false]{filehooks}
46 \DeclareStringOption[false]{autokw}[all]
47
48 \ExecuteOptions{old}
49 \ProcessKeyvalOptions{svn-multi}
50
51 Enable dependent options:
52 \def\svn@depooption#1{%
53 \csname if@svnmulti@#1\endcsname\else
54 \message{svn-multi: Required option '#1' enabled.}%
55 \csname @svnmulti@#1true\endcsname
56 \fi
57 }
58
59 \if@svnmulti@groups
60 \svn@depooption{filehooks}
61 \fi
62
63 \if@svnmulti@external
64 \svn@depooption{filehooks}
65 \fi
66
67 \if@svnmulti@subgroups
68 \svn@depooption{groups}
69 \fi
70
71 \if@svnmulti@graphics
72 \svn@depooption{pgfimages}
73 \fi
74
75 \if@svnmulti@table
76 \svn@depooption{table}
77 \fi
78
79 \if@svnmulti@filehooks
80 \svn@depooption{autokw}
81 \fi
82
83 \if@svnmulti@autoload
84 \svn@depooption{true}
85 \fi
86
87 \if@svnmulti@all
88 \svn@depooption{all}
89 \fi
90
91 \if@svnmulti@all
92 \message{svn-multi: All options enabled.}%
93 \fi
94
95 \if@svnmulti@all
96 \message{svn-multi: All options enabled.}%
97 \fi
98
99 \if@svnmulti@all
100 \message{svn-multi: All options enabled.}%
101 \fi

```

```

66 \fi
67 \if@svnmulti@graphics
68   \svn@depooption{external}
69   \svn@depooption{autoload}
70   \svn@depooption{filehooks}
71 \fi
72 \if@svnmulti@pgfimages
73   \svn@depooption{external}
74   \svn@depooption{autoload}
75   \svn@depooption{filehooks}
76 \fi
77 \if@svnmulti@autoload
78   \svn@depooption{external}
79   \svn@depooption{filehooks}
80 \fi
81 \if@svnmulti@table
82   \svn@depooption{groups}
83   \svn@depooption{filehooks}
84 \fi

```

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

```

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

```

Set **autokw** modes:

```

106 \def\svn@temp{true}
107 \ifx\@svnmulti@autokw\svn@temp
108   \@svnmulti@autokwtrue
109   \@svnmulti@autokwalltrue
110   \svn@depooption{filehooks}
111 \fi
112 \def\svn@temp{all}

```

```

113 \ifx\@svnmulti@autokw\svn@temp
114   \@svnmulti@autokwtrue
115   \@svnmulti@autokwalltrue
116   \svn@depooption{filehooks}
117 \fi
118 \def\svn@temp{ext}
119 \ifx\@svnmulti@autokw\svn@temp
120   \@svnmulti@autokwtrue
121   \@svnmulti@autokwallfalse
122 \fi
123 \def\svn@temp{false}
124 \ifx\@svnmulti@autokw\svn@temp
125   \@svnmulti@autokwfalse
126   \@svnmulti@autokwallfalse
127 \fi

    General switch if any graphic option is enabled:

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

```

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

```

### **\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.

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

#### **\svn@ifvalidrev**

**#1:** macro name

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

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

#### **\svn@ifeof**

**#1:** input file handle

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

```
177 \def\svn@ifeof#1{%
178   \ifeof#1%
179     \expandafter\@firstoftwo
180   \else
181     \expandafter\@secondoftwo
182   \fi
```

```
183 }
```

### **\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.

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

### **\svn@input**

**#1:** file name/path

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

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

### **\svn@inputsvx**

**#1:** file name/path without extension

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

```
199 \def\svn@inputsvx#1{%
200   \svn@pushfilestack
201   \svn@input{#1.svx}%
202   \svn@popfilestack
203 }
```

## **7.3 Definition of init values**

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

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

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

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

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

```

204 % Init values
205 \def\svnrev{-1}          \def\@svn@rev{-1}
206 \def\svndate{}          \def\@svn@date{}
207 \def\svnauthor{}        \def\@svn@author{}
208 \def\svnyear{0000}      \def\@svn@year{0000}
209 \def\svnmonth{00}       \def\@svn@month{00}
210 \def\svnday{00}         \def\@svn@day{00}
211 \def\svnhour{00}        \def\@svn@hour{00}
212 \def\svnminute{00}     \def\@svn@minute{00}
213 \def\svnsecond{00}     \def\@svn@second{00}
214 \def\svntimezonehour{+00} \def\@svn@timezonehour{+00}
215 \def\svntimezoneminute{00} \def\@svn@timezoneminute{00}
216 \def\svnmainurl{NOT SET} \def\svnmainfilename{NOT SET}
217 \def\svnurl{} \def\svnfname{}
218 \def\svn@temp{}
219
220 \def\svn@pg{} \def\svn@g{} \def\svn@cg{\svn@g} \def\svn@rg{\svn@pg}
221 \let\@svng@@files\relax
222
223 \def\svn@initfile{%
224   \gdef\svnfilerev{-1}%
225   \gdef\svnfiledate{}%
226   \gdef\svnfileauthor{}%
227   \gdef\svnfileyear{0000}%
228   \gdef\svnfilemonth{00}%
229   \gdef\svnfileday{00}%
230   \gdef\svnfilehour{00}%
231   \gdef\svnfileminute{00}%
232   \gdef\svnfilesecond{00}%
233   \gdef\svnfiletimezonehour{+00}%
234   \gdef\svnfiletimezoneminute{00}%
235   \gdef\svnfileurl{}%
236   \gdef\svnfilefname{}%
237   \gdef\svnfilemdir{}%
238 }
239 \svn@initfile

```

## 7.4 Auto-Keywords

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

```
240 \begingroup
241 \@makeother\^^L
242 \if@svnmulti@autokw
243 \gdef\svne@ff{^^L}
244 \fi
245 \endgroup

246 \if@svnmulti@autokw
247 \newread\svne@read
```

### **\svne@catcodes**

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

```
248 \newcommand*{\svne@catcodes}{%
249   \let\do\@makeother
250   \endlinechar=-1
251   \dospecials
252   \do\~ \do\: \do\. \do\^^L
253 }
```

### **\svne@readline**

#1: macro

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

```
254 \def\svne@readline#1{%
255   \ifeof\svne@read
256     \def#1{}%
257   \else
258     \read\svne@read to #1\relax
259   \fi
260 }
```

### **\svne@gobblerest**

Gobbles the rest of the current entry.

```
261 \def\svne@gobblerest{%
262   \ifeof\svne@read
263     \let\next\relax
264   \else
265     \read\svne@read to \svn@temp
266     \ifx\svn@temp\svne@ff
267       \let\next\relax
268     \else
269       \let\next\svne@gobblerest
270     \fi
271   \fi
```

```

272 \next
273 }

```

#### **\svne@endread**

Stops the reading process of the entries file.

```

274 \def\svne@endread{%
275   \closein\svne@read
276 }

```

#### **\svne@parseentriesfile**

#1: file path

```

277 \newcommand*{\svne@parseentriesfile}[1]{%
278   \begingroup
279   \let\next\relax

```

Open the format file to read the version number:

```

280   \openin\svne@read=#1format\relax
281   \ifeof\svne@read\else
282     \svne@readline\svne@version
283   \closein\svne@read

```

Check the format version:

```

284   \ifnum\svne@version>7\relax

```

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

```

285   \openin\svne@read=#1entries\relax
286   \ifeof\svne@read\else
287     \svne@catcodes
288     \svne@readline\svne@version

```

Check the version and call the parse macros if OK:

```

289     \ifnum\svne@version>7\relax
290       \def\next{\svne@parsedirentury%
291         \svne@parseentries}
292     \else
293       \closein\svne@read
294     \fi
295   \fi
296   \fi
297   \fi
298   \next
299 \endgroup
300 }

```

#### **\svne@parsedirentury**

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

```

301 \newcommand*{\svne@parsedirentury}{%
302   \svne@readline\svne@name
303   \svne@readline\svne@kind

```

```

304 \svn@ifempty{\svne@name}%
305   {\svn@ifequal{\svne@kind}{dir}%
306     {%
307       {\svne@readline\svn@temp}%
308       \svne@readline\svne@baseurl
309       \svne@gobblertest
310     }{}%
311   }{}%
312 }

```

#### **\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.

```

313 \begingroup
314
315 \makeother\~
316 \makeother\:
317 \makeother\.
318
319 \gdef\svne@scandate#1{%
320   \expandafter\svne@scandate@#1\relax
321   0000-00-00T00:00:00.00000Z\relax\relax
322 }
323
324 \gdef\svne@scandate@#1-#2-#3T#4:#5:#6.#7\relax#8\relax{%
325   \gdef\svnfileyear{#1}%
326   \gdef\svnfilemonth{#2}%
327   \gdef\svnfileday{#3}%
328   \gdef\svnfilehour{#4}%
329   \gdef\svnfileminute{#5}%
330   \gdef\svnfilesecond{#6}%
331   \gdef\svnfiletimezonehour{+00}%
332   \gdef\svnfiletimezoneminute{00}%
333   \gdef\svnfiledate{#1-#2-#3 #4:#5:#6Z}%
334   \def\svne@date{#1-#2-#3 #4:#5:#6Z}%
335 }
336
337 \endgroup

```

#### **\svne@parseentries**

```

338 \newcommand*{\svne@parseentries}{%
339   \svn@ifeof{\svne@read}%
340   {}%
341   {%
342     \svne@readline\svne@name
343     \@onelevel@sanitize\svne@name

```

```

344 \svn@ifeof{\svne@read}%
345 {}%
346 {%
347 \svne@readline\svne@kind
348 \svn@ifequal{\svne@kind}{file}%
349 {%
350 \svne@readline\svn@temp
351 \svne@readline\svn@temp
352 \svne@readline\svn@temp
353 \svne@readline\svn@temp
354 \svne@readline\svn@temp
355 \svne@readline\svn@temp
356 \svne@readline\svne@date
357 \svne@readline\svne@rev
358 \svne@readline\svne@author
359 %\@onelevel@sanitize\svne@date
360 \svne@scandate{\svne@date}%
361 \edef\svne@url{\svne@baseurl/\svne@name}%
362 \svne@handleentry
363 }{}%
364 \svne@gobblertest
365 \svne@parseentries
366 }%
367 }%
368 }

```

### **\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 `\svnidl@long`. Then the `\svne@endread` macro is used to stop the file reading.

```

369 \def\svne@handleentry{%
370 \svn@ifequal{\svne@name}{\svnfilename}%
371 {%
372 \svnkwdef{Filename}{\svne@name}%
373 \svnkwdef{Date}{\svne@date}%
374 \svnkwdef{Revision}{\svne@rev}%
375 \svnkwdef{Author}{\svne@author}%
376 \svnkwdef{HeadURL}{\svne@url}%
377 \@svn@updateid{\svne@rev}{\svne@date}{\svne@author}{\svne@url}%
378 \svne@endread
379 }{}%
380 }%

```

### **\svnegetfile**

#1: file path

```

381 \def\svngetfile#1{%
382   \begingroup
383     \svn@getfilename{#1}%
384     \edef\svnfilefname{\svnfilefname}%
385     \@onelevel@sanitize\svnfilefname
386     \svn@parseentriesfile{\svnfiledir .svn/}%
387     \svn@parseentriesfile{\svnfiledir _svn/}%
388   \endgroup
389 }

```

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

```

390 \if@svnmulti@autokwall
391 \AtBeginDocument{%
392   \svngetfile{\jobname.\svn@mainext}%
393 }
394 \fi
395 \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.

```

396 \def\svntimezone{\svntimezonehour\svntimezoneminute\svn@gobblezeros}
397 \def\svnfiletimezone{\svnfiletimezonehour\svnfiletimezoneminute\svn@gobblezeros}
398 \def\svncgtimezone{\svncgtimezonehour\svncgtimezoneminute}

```

**\svn@gobblezeros**

**\svn@gobblezeros@**

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

```

399 \def\svn@gobblezeros{%
400   \futurelet\svn@nextchar\svn@gobblezeros@
401 }
402 \def\svn@gobblezeros@{%
403   \let\@tempa=\relax
404   \def\@tempb{0}%
405   \ifx0\svn@nextchar
406     \let\@tempa=\@gobbletwo
407   \fi
408   \@tempa
409 }

```



`\svntime`

`\svnfiletime`

`\svncgtime`

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

```
410 \def\svntime{\svnhour:\svnminute:\svnsecond}
411 \def\svnfiletime{\svnfilehour:\svnfileminute:\svnfilesecond}
412 \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`

```
413 \newcommand*{\svntoday}{%
414   \begingroup
415     \year\svnyear \month\svnmonth \day\svnday
416     \relax \today
417   \endgroup
418 }
```

`\svnfiletoday`

```
419 \newcommand*{\svnfiletoday}{%
420   \begingroup
421     \year\svnfileyear \month\svnfilemonth \day\svnfileday
422     \relax \today
423   \endgroup
424 }
```

`\svncgtoday`

```
425 \newcommand*{\svncgtoday}{%
426   \@ifundefined{svng@\svn@cgs @year}{??}{%
427     \begingroup
428       \year\svncgyear \month\svncgmonth \day\svncgday
429       \relax \today
430     \endgroup
431   }%
432 }
```

## 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`.

```

433 \newcommand*{\svnId}{%
434   \@svnidswtrue
435   \svnkwsave
436 }
437 \newif\if@svnidsw
438 \@svnidswfalse

```

#### **\svn@scanId**

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

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

```

439 \def\svn@scanId#1 #2 #3 #4 #5\relax{%
440   \@svn@scandate{#3 #4}%
441   \@svn@updateid{#2}{#3 #4}{#5}{#1}%
442   \svnkwnode{Filename}{#1}%
443   \svnkwnode{Date}{#3 #4}%
444   \svnkwnode{Revision}{#2}%
445   \svnkwnode{Author}{#5}%
446 }

```

#### **\@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.

```

447 \def\@svn@updateid#1#2#3#4{%
448   \xdef\svnfilerev{#1}%
449   \xdef\svnfiledate{#2}%
450   \xdef\svnfileauthor{#3}%
451   \xdef\svnfileurl{#4}%
452   \svn@getfilename\svnfileurl%

```

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

```

453   \ifx\svnfilerev\empty\else
454     \ifnum\@svn@rev<\svnfilerev
455       \xdef\@svn@rev{\svnfilerev}%

```

```

456 \xdef\@svn@date{\svnfiledate}%
457 \xdef\@svn@author{\svnfileauthor}%
458 \xdef\@svn@year{\svnfileyear}%
459 \xdef\@svn@month{\svnfilemonth}%
460 \xdef\@svn@day{\svnfileday}%
461 \xdef\@svn@hour{\svnfilehour}%
462 \xdef\@svn@minute{\svnfileminute}%
463 \xdef\@svn@second{\svnfilesecond}%
464 \xdef\@svn@timezonehour{\svnfiletimezonehour}%
465 \xdef\@svn@timezoneminute{\svnfiletimezoneminute}%
466 \xdef\@svn@url{\svnfileurl}%
467 \xdef\@svn@fname{\svnfilefname}%
468 \fi
469
470 \if@svnmulti@groups
471 \ifx\svn@g\empty\else
472 \svn@updategroup{\svn@g}%
473 \fi
474 \if@svnmulti@subgroups
475 \ifsvnsubgroups
476 \svn@updategroup{\svn@filedir\svn@filebase}%
477 \fi
478 \fi
479 \fi
480 \fi
481 }
482
483 \def\@svncg@save#1#2{%
484 \expandafter\xdef\csname @svng@\svn@g @#1\endcsname{#2}%
485 }
486

```

### 7.7.2 Long Id

#### `\svnidlong`

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

```

487 \newcommand{\svnidlong}{%
488 \svnkundef{HeadURL}{}%
489 \svnkundef{LastChangedDate}{}%
490 \svnkundef{LastChangedRevision}{0}%
491 \svnkundef{LastChangedBy}{}%

```

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

```

492 \begingroup
493 \if@svnmulti@verbatim
494 \svn@catcodes
495 \catcode'\{=12

```

```

496     \catcode'\}=12
497   \else
498     \def\svnidl@readargsfull{\svnidl@readargs}%
499   \fi
500   \svnidl@readargsfull
501 }

```

#### `\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.

```

502 \if@svnmulti@verbatim
503 \def\svn@catcodes{%
504   \let\do\@makeother
505   \dospecials
506   \catcode'\^M9
507   \catcode'\ 10
508   \catcode'\{1
509   \catcode'\}2
510 }
511 \else
512   \def\svn@catcodes{}
513 \fi

```

#### `\svnidl@readargsfull`

```

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

```

Reads all four arguments of `\svnidl` and passes them to `\svnidl@readargs`. The normal argument braces are changed to category “other” and put into the macros parameter text to remove all code between them. This is done to avoid problems with comments direct after one of the arguments. Because the braces are now non-special the parentheses are made to a local replacement.

```

514 \if@svnmulti@verbatim
515 \begingroup
516 \catcode'\{=12\catcode'\}=12
517 \catcode'\(=1\catcode'\)=2
518 \gdef\svnidl@readargsfull#1{#2}#3{#4}#5{#6}#7{#8}{%
519   \svnidl@readargs(#2)(#4)(#6)(#8)%
520 }
521 \endgroup
522 \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`.

```
523 \def\svnidlong@readargs#1#2#3#4{%  
524     \svnkwsave@read #1\relax  
525     \svnkwsave@read #2\relax  
526     \svnkwsave@read #3\relax  
527     \svnkwsave@read #4\relax  
528 \endgroup
```

Now the update macros for date and id are called.

```
529 \ifx\svnkWLastChangedDate\empty\else  
530     \@svn@scanlongdate{\svnkWLastChangedDate}%  
531 \fi  
532 \@svn@updateid{\svnkW{LastChangedRevision}}{\svnkW{LastChangedDate}}%  
533 {\svnkW{LastChangedBy}}{\svnkW{HeadURL}}%  
534 \ignorespaces  
535 }%
```

## 7.8 KeyWord Macros

### `\svnkwsave`

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

```
536 \def\svnkwsave{%  
537     \begingroup  
538     \svn@catcodes  
539     \svnkwsave@readargs  
540 }
```

### `\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.

```
541 \gdef\svnkwsave@readargs#1{%  
542     \svnkwsave@read#1\relax  
543 \endgroup  
544 \if@svnidsw  
545     \ifx\svnkWId\empty\else  
546         \expandafter  
547         \svn@scanId\svnkWId\relax  
548         \@svnidswfalse  
549     \fi
```

```

550 \fi
551 \ignorespaces
552 }

```

#### **\svnkwsave@read**

#1: keyword line without surrounding \$ \$  
 Reads the full keyword and strips the dollars.

```

553 \begingroup
554 \if@svnmulti@verbatim
555 \catcode'\$=12
556 \fi
557 \gdef\svnkwsave@read $#1$\relax{%
558   \svn@checkcolon#1:\relax
559 }
560 \endgroup

```

#### **\svnkwsave@parse**

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

```

561 \begingroup
562 \catcode'\$=11
563 \gdef\svnkwsave@parse$#1:#2${%
564   \expandafter\xdef\csname svnkw#1\endcsname{#2}%
565 }%
566 \endgroup

```

#### **\svnkwdef**

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

```

567 \newcommand{\svnkwdef}[2]{%
568   \@ifundefined{svnkwdef@#1}%

```

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

```

569   {\svnkwdef@{#1}{#2}}%

```

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

```

570   {\csname svnkwdef@#1\endcsname{#2}}%
571 }

```

#### **\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.

```

572 \newcommand{\svnkwdef@}[2]{%
573   \expandafter\xdef\csname svnkw#1\endcsname{#2}%
574 }

```

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

`\svnkwdef@Rev`

`\svnkwdef@Author`

`\svnkwdef@Date`

`\svnkwdef@URL`

#1: value

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

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

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

```

575 \def\svnkwdef@Rev#1{%
576   \svn@ifempty{#1}%
577     {\svnkwdef@{Rev}{0}}%
578     {\svnkwdef@{Rev}{#1}}%
579 }
580 \def\svnkwdef@Author#1{\svnkwdef@{Author}{#1}}
581 \def\svnkwdef@Date#1{\svnkwdef@{Date}{#1}}
582 \def\svnkwdef@URL#1{\svnkwdef@{HeadURL}{#1}}

```

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

```

583 \let\svnkwdef@Revision=\svnkwdef@Rev
584 \let\svnkwdef@LastChangedRevision=\svnkwdef@Rev
585 \let\svnkwdef@LastChangedBy=\svnkwdef@Author
586 \let\svnkwdef@LastChangedAt=\svnkwdef@Date

```

and then for reading.

```

587 \def\svnkwRevision{\svnkwRev}
588 \def\svnkwLastChangedRevision{\svnkwRev}
589 \def\svnkwLastChangedBy{\svnkwAuthor}
590 \def\svnkwLastChangedAt{\svnkwDate}
591 \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.

```

592 \svnkwdef{Rev}{0}
593 \svnkwdef{Date}{}
594 \svnkwdef{Author}{}
595 \svnkwdef{Filename}{}
596 \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.

```
597 \newcommand{\svnkw}[1]{%
598   \ifundefined{svnkw#1}%
599     {\PackageWarning{svn-multi}{SVN keyword '#1' not defined (typo?)}}%
600     {\csname svnkw#1\endcsname}%
601 }
```

## 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 L<sup>A</sup>T<sub>E</sub>X 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`.

```
602 \def\svn@checkcolon#1:#2\relax{%
603   \svn@ifempty{#2}%
604     {\svnkwdef{#1}{}}%
605     {\svn@stripcolon#2\relax\svnkwdef{#1}{\svn@value}}%
606 }
```

### `\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@stripspace` is called to strip the spaces.

```
607 \def\svn@stripcolon#1:\relax{%
608   \svn@ifempty{#1}%
609     {\gdef\svn@value{}}%
610     {\svn@ifequal{#1}{ }%
611       {\gdef\svn@value{}}%
612       {\svn@stripspace#1\relax\relax}%
613     }
```



```

613     }%
614 }

```

### **\svn@stripspace**

#1: first character  
 #2: rest of string

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

```

615 \def\svn@stripspace#1#2\relax{%
616   \svn@ifequal{#1}{ }%
617   {\gdef\svn@value{#2}}%
618   {\svn@striptrailingspace#1#2\relax}%
619 }

```

### **\svn@striptrailingspace**

#1: string

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

```

620 \def\svn@striptrailingspace#1 \relax{%
621   \gdef\svn@value{#1}%
622 }

```

### **\svn@gdefverb**

#1: macro

```

623 \def\svn@gdefverb#1{%
624   \begingroup
625   \def\svn@temp{#1}%
626   \begingroup
627   \if@svnmulti@verbatim
628     \svn@catcodes
629   \fi
630   \svn@gdefverb@
631 }

```

### **\svn@defverb@**

#1: verbatim stuff

```

632 \def\svn@gdefverb@#1{%
633   \endgroup
634   \expandafter\gdef\svn@temp{#1}%
635   \endgroup
636 }

```

### **\svn@namegdefverb**

#1: macro name

```

637 \def\svn@namegdefverb#1{%
638   \begingroup

```

```

639 \expandafter\def
640 \expandafter\svn@temp
641 \expandafter{\csname #1\endcsname}%
642 \beginingroup
643 \if@svnmulti@verbatim
644 \svn@catcodes
645 \fi
646 \svn@gdefverb@
647 }

```

## 7.10 Date Macros

### \@svn@scandate

#1: date

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

```

648 \def\@svn@scandate#1{\@svn@scandate@#1\relax}
649
650 \def\@svn@scandate@#1-#2-#3 #4:#5:#6#7#8\relax{%
651 \gdef\svnfileyear{#1}%
652 \gdef\svnfilemonth{#2}%
653 \gdef\svnfileday{#3}%
654 \gdef\svnfilehour{#4}%
655 \gdef\svnfileminute{#5}%
656 \gdef\svnfilesecond{#6#7}%
657 \gdef\svnfiletimezonehour{+00}%
658 \gdef\svnfiletimezoneminute{00}% #8 always 'Z' for Zulu-time (UTC)
659 }

```

### \@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.

```

660 \def\@svn@scanlongdate#1{\expandafter\@svn@scanlongdate@#1\relax}
661 %
662 \def\@svn@scanlongdate@#1-#2-#3 #4:#5:#6 #7 #8\relax{%
663 \gdef\svnfileyear{#1}%
664 \gdef\svnfilemonth{#2}%
665 \gdef\svnfileday{#3}%
666 \gdef\svnfilehour{#4}%
667 \gdef\svnfileminute{#5}%
668 \gdef\svnfilesecond{#6}%

```

```

669 \@svn@parsetimezone#7\relax%
670 }

```

#### **\@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.

```

671 \def\@svn@parsetimezone#1#2#3#4#5\relax{%
672   \gdef\svnfiletimezonehour{#1#2#3}%
673   \gdef\svnfiletimezoneminute{#4#5}%
674 }

```

#### **\svnpdfdate**

Returns date in a format needed for \pdfinfo.

```

675 \def\svnpdfdate{%
676   \svnyear\svnmonth\svnday
677   \svnhour\svnminute\svnsecond\svntimezonehour'\svntimezoneminute'%
678 }

```

## **7.11 Mainfile Makros**

#### **\svnsetmainfile**

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

```

679 \newcommand{\svnsetmainfile}{%
680   \xdef\svnmainurl{\svnfileurl}%
681   \xdef\svnmainfilename{\svnfilename}%
682 }
683 \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.

```

684 \newcommand{\svnRegisterAuthor}[2]{%
685   \expandafter\def\csname svn@author@#1\endcsname{#2}%
686 }

```

**\svnFullAuthor**

**\svnFullAuthor\***

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

```
687 \newcommand{\svnFullAuthor}{%  
688   \ifnextchar{*}%  
689     {\svnFullAuthor@star}%  
690     {\svnFullAuthor@normal}%  
691 }%
```

**\svnFullAuthor@star**

#1: username

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

```
692 \def\svnFullAuthor@star*#1{%  
693   \edef\svn@temp{#1}%  
694   \svnFullAuthor@{\svn@temp}{~(\svn@temp)}%  
695 }%
```

**\svnFullAuthor@normal**

#1: username

```
696 \def\svnFullAuthor@normal#1{%  
697   \edef\svn@temp{#1}%  
698   \svnFullAuthor@{\svn@temp}{}%  
699 }%
```

**\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.

```
700 \def\svnFullAuthor@#1#2{%  
701   \@ifundefined{svn@author@#1}%  
702     {#1}%  
703     {\csname svn@author@#1\endcsname #2}%  
704 }
```

**\svnRegisterRevision**

#1: revision number

#2: tag name

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

```
705 \newcommand{\svnRegisterRevision}[2]{%  
706   \expandafter\def\csname svn@revision@#1\endcsname{#2}%  
707 }
```

**\svnFullRevision**

**\svnFullRevision\***

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

```
708 \newcommand{\svnFullRevision}{%  
709   \@ifnextchar{*}%  
710   {\svnFullRevision@star}%  
711   {\svnFullRevision@normal}%  
712 }
```

**\svnFullRevision@star**

#1: revision number

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

```
713 \def\svnFullRevision@star*#1{%  
714   \edef\svn@temp{#1}%  
715   \svnFullRevision@\svn@temp}{~(r\svn@temp)}%  
716 }
```

**\svnFullRevision@normal**

#1: revision number

```
717 \def\svnFullRevision@normal#1{%  
718   \edef\svn@temp{#1}%  
719   \svnFullRevision@\svn@temp}{}%  
720 }
```

**\svnFullRevision@**

#1: revision number

#2: previous defined trailing string

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

```
721 \def\svnFullRevision@#1#2{%  
722   \@ifundefined{svn@revision@#1}%  
723   {Revision #1}%  
724   {\csname svn@revision@#1\endcsname #2}%  
725 }
```

## 7.13 Input File Name

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

```
726 \if@svnmulti@filehooks
```

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

```
727 \RequirePackage{fink}[2008/02/27]  
728 \begingroup
```

```

729 \def\svn@finkerror{%
730 \PackageError{svn-multi}{Your installed version of the 'fink' package does not
731 provide the needed macros. It is either too old or too new.
732 Try a different version, e.g. v2.1.1 from 2008/02/27}{}}%
733 \let\svn@finkerror\relax
734 }
735 \ifundefined{finkpath}{\svn@finkerror}{}%
736 \ifundefined{finkdir}{\svn@finkerror}{}%
737 \ifundefined{finkbase}{\svn@finkerror}{}%
738 \ifundefined{fink@prepare}{\svn@finkerror}{}%
739 \ifundefined{fink@restore}{\svn@finkerror}{}%
740 \ifundefined{fink@maindir}{\svn@finkerror}{}%
741 \ifundefined{fink@mainext}{\svn@finkerror}{}%
742 \endgroup

```

### **\svn@removedotslash**

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

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

```

743 \def\svn@removedotslash#1{%
744   \def\svn@removedotslash##1##2##3\relax{%
745     \svn@ifequal{.}{##1##2}%
746     {\edef#1{##3}\def\next{\svn@removedotslash###\empty\empty\empty\relax}}%
747     {\edef#1{##1##2##3}\let\next\relax}%
748     \next
749   }%
750   \expandafter\svn@removedotslash@#1\empty\empty\empty\relax
751 }

```

Init values for file name macros.

```

752 \let\svn@mainext\fink@mainext
753 \let\svn@maindir\fink@maindir
754 \svn@removedotslash\svn@maindir
755 \edef\svn@filebase{\jobname}%
756 \edef\svn@fileext{\svn@mainext}%
757 \edef\svn@filedir{\svn@maindir}%

```

Filename and -path are build using the other macros:

```

758 \def\svn@filename{\fink@file\svn@filebase\svn@fileext}%
759 \def\svn@filepath{\svn@filedir\svn@filename}%

```

### **\svnmulti@begininputfilehook**

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

```

760 \def\svnmulti@begininputfilehook{}
761 \message{Package svn-multi: patching macro '\string\fink@prepare' from the
762 'fink' package!}%
763 \let\svnmulti@fink@prepare\fink@prepare

```

```

764 \renewcommand*{\fink@prepare}[1]{%
765   \svnmulti@fink@prepare{#1}%
766   \if@svnmulti@groups
767     \svn@ifequal{\svn@filepath}{\jobname.\svn@mainext}%
768     {\global\let\svn@pg\svn@g}%
769     {\xdef\svn@pg{\svn@filedir\svn@filebase}}}%
770   \fi
771   \xdef\svn@filebase{\fink@nextbase}%
772   \xdef\svn@fileext{\fink@nextext}%
773   \xdef\svn@filedir{\fink@nextdir}%
774   \svn@removedotslash\svn@filedir
775   \svnmulti@begininputfilehook
776 }%

```

#### `\svnmulti@endinputfilehook`

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

```

777 \def\svnmulti@endinputfilehook{}
778 \message{Package svn-multi: patching macro '\string\fink@restore' from the
779 'fink' package!}%
780 \let\svnmulti@fink@restore\fink@restore
781 \def\fink@restore#1{%
782   \svnmulti@endinputfilehook
783   \svnmulti@fink@restore{#1}%
784   \xdef\svn@filebase{\finkbase}%
785   \xdef\svn@fileext{\finkext}%
786   \xdef\svn@filedir{\finkdir}%
787   \svn@removedotslash\svn@filedir
788 }%

```

#### `\svnmulti@atbegininputfile`

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

```

789 \def\svnmulti@atbegininputfile{%
790   \g@addto@macro\svnmulti@begininputfilehook
791 }

```

#### `\svnmulti@atendinputfile`

This macro adds the argument to the *begin* of the `\svnmulti@endinputfilehook`. This ensures that code added first is more at the end than code added later. The code below was adapted from the definition of the  $\text{\LaTeX}2\text{e}$  macro `\g@addto@macro` which was used above.

```

792 \long\def\svnmulti@atendinputfile#1{%
793   \begingroup
794     \@temptokena\expandafter{\svnmulti@endinputfilehook}%
795     \toks@{#1}%
796     \xdef\svnmulti@endinputfilehook{\the\toks@\the\@temptokena}%
797   \endgroup
798 }

```

```

799 \def\svn@filestack{}}
800
801 \def\svn@pushfilestack{%
802   \xdef\svn@filestack{%
803     {\svnfilerev}%
804     {\svnfiledate}%
805     {\svnfileauthor}%
806     {\svnfileyear}%
807     {\svnfilemonth}%
808     {\svnfileday}%
809     {\svnfilehour}%
810     {\svnfileminute}%
811     {\svnfilesecond}%
812     {\svnfiletimezonehour}%
813     {\svnfiletimezoneminute}%
814     {\svnfileurl}%
815     {\svnfilefname}%
816     {\svn@g}%
817     {\svn@pg}%
818   }\svn@filestack}%
819 }
820
821 \def\svn@restorefilekws#1#2\relax{%
822   \svn@restorefilekws@#1\empty
823   \empty \empty \empty \empty
824   \empty \empty \empty \empty
825   \empty \empty \empty \empty
826   \svn@ifempty{#2}%
827     {\gdef\svn@filestack{}}}%
828     {\gdef\svn@filestack{#2}}}%
829 }
830 \def\svn@restorefilekws@#1#2#3#4#5#6#7#8#9{%
831   \gdef\svnfilerev{#1}%
832   \gdef\svnfiledate{#2}%
833   \gdef\svnfileauthor{#3}%
834   \gdef\svnfileyear{#4}%
835   \gdef\svnfilemonth{#5}%
836   \gdef\svnfileday{#6}%
837   \gdef\svnfilehour{#7}%
838   \gdef\svnfileminute{#8}%
839   \gdef\svnfilesecond{#9}%
840   \svn@restorefilekws@@
841 }
842
843 \def\svn@restorefilekws@@#1#2#3#4#5#6{%
844   \gdef\svnfiletimezonehour{#1}%
845   \gdef\svnfiletimezoneminute{#2}%
846   \gdef\svnfileurl{#3}%
847   \gdef\svnfilefname{#4}%
848   \gdef\svn@g{#5}%

```



```

849 \gdef\svn@pg{#6}%
850 }
851
852 \def\svn@popfilestack{%
853   \ifx\svn@filestack\empty
854     \PackageWarning{svn-multi}{Underflow of file keyword stack!}%
855   \else
856     \svn@ifequal{\svn@filestack}{\{}}%
857     {\PackageWarning{svn-multi}{Underflow of file keyword stack!}}%
858     {\expandafter\svn@restorefilekws\svn@filestack\relax}%
859   \fi
860 }
861
862 \svnmulti@atbegininputfile{%
863   \svn@pushfilestack
864 }
865
866 \svnmulti@atendinputfile{%
867   \svn@popfilestack
868 }%
869 %
870 \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`.

```

871 \if@svnmulti@groups
872 \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.

```

873 \def\svngroup#1{%
874   \svn@ifequal{#1}{*}%
875   {\PackageError{svn-multi}%
876     {The group name '*' is invalid for '\string\svngroup'}{\{}}%
877   }{}%
878   \xdef\svn@g{#1}%
879   \let\svn@pg\svn@g
880   \ifx\svn@g\empty\else%

```

Only initialise the group at first usage:

```

881 \expandafter
882 \ifx\csname @svng@\svn@g @rev\endcsname\relax%
883 \svn@initgroup{\svn@g}%

```

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

```

884 \ifx\svn@glist\empty
885 \xdef\svn@glist{#1}%
886 \else
887 \xdef\svn@glist{\svn@glist,#1}%
888 \fi
889 \fi
890 \fi
891 }

```

### **\thesvngroup**

Returns the current group name to the user.

```

892 \def\thesvngroup{\svn@g}

```

### **\svnsetcg**

#1: group name

Defines \svn@cg to the given argument or to \svn@g if the argument was ‘\*’.

```

893 \def\svnsetcg#1{%
894 \svn@ifequal{#1}{*}%
895 {\def\svn@cg{\svn@g}}%
896 {\def\svn@cg{#1}}%
897 }

```

### **\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.

```

898 \def\svncg@def#1{%
899 \expandafter
900 \def\csname svncg#1\endcsname{%
901 \@ifundefined{svng@\svn@cg @#1}{??}{%
902 \csname svng@\svn@cg @#1\endcsname}%
903 }%
904 }

```

### **\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.

```

905 \@for\@tempa:=%
906   rev,author,date,year,month,day,hour,minute,second,%
907   timezonehour,timezoneminute,url,fname%
908 \do{%
909   \expandafter\svncg@def\expandafter{\@tempa}%
910 }
```

`\thesvncg`

Simply return the internal macro.

```

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

`\svng`

#1: group name

#2: keyword name

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

```

912 \def\svng#1#2{%
913   \ifundefined{svng@\svn@temp @#2}%
914     {??}%
915     {\csname svng@\svn@temp @#2\endcsname}%
916 }
```

`\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.

```

917 \def\svn@addfiletogroup#1#2{%
918   \expandafter
919   \ifx\csname @svng@#2@files@#1\endcsname\relax%
920     \expandafter\gdef\csname @svng@#2@files@#1\endcsname{1}%
921     %
922     \ifundefined{@svng@#2@files}%
923       {\expandafter\xdef\csname @svng@#2@files\endcsname{#1}}%
924       {\expandafter\xdef\csname @svng@#2@files\endcsname{%
925         \csname @svng@#2@files\endcsname,#1%
```

```

926     }%
927     }%
928   \fi
929 }

```

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}`).

```

930 \svnmulti@atbegininputfile{%
931   \svn@ifequal{\svn@filepath}{\svn@maindir\jobname.\svn@mainext}%
932   }%
933   {\svn@initfile}%
934   \svn@ifequal{\svn@fileext}{\svn@mainext}%
935   {\svn@addfiletogroup{\svn@filedir\svn@filebase}{\svn@pg}}}%
936   \svn@ifequal{\svn@fileext}{sty}%
937   {\svn@addfiletogroup{\svn@filedir\svn@filebase}{\svn@pg}}}%
938   \svn@ifequal{\svn@fileext}{cls}%
939   {\svn@addfiletogroup{\svn@filedir\svn@filebase}{\svn@pg}}}%
940   \svn@addfiletogroup{\svn@filepath}{\svn@filedir\svn@filebase}%
941 }

```

### `\svn@writegroup`

#1: group name

Writes group to `\svn@write` file.

```

942 \def\svn@writegroup#1{%
943   \def\svn@writekw##1{%
944     \immediate\write\svn@write{%
945       \noexpand\@namedef{svng@#1@##1}{\csname @svng@#1@##1\endcsname}%
946     }%
947   }%
948   \svn@writekw{rev}%
949   \svn@writekw{date}%
950   \svn@writekw{author}%
951   \svn@writekw{year}%
952   \svn@writekw{month}%
953   \svn@writekw{day}%
954   \svn@writekw{hour}%
955   \svn@writekw{minute}%
956   \svn@writekw{second}%
957   \svn@writekw{timezonehour}%
958   \svn@writekw{timezoneminute}%
959   \@ifundefined{svng@#1@files}{\csname @svng@#1@files\endcsname}%
960   \immediate\write\svn@write{%
961     \noexpand
962     \svn@namegdefverb{svng@#1@files}{\csname @svng@#1@files\endcsname}%
963   }%
964   }%
965   \immediate\write\svn@write{%

```

```

966 \noexpand
967 \svn@namegdefverb{svng@#1@url}{\csname @svng@#1@url\endcsname}^^J%
968 \noexpand
969 \svn@namegdefverb{svng@#1@fname}{\csname @svng@#1@fname\endcsname}^^J%
970 }%
971 }

```

### **\svn@writeallgroups**

#1: macro holding a list of groups

```

972 \def\svn@writeallgroups#1{%
973   \beginngroup
974   \ifx\relax#1\relax\else
975     \@for\svn@temp:=#1\do{%
976       \svn@ifvalidrev{svng@\svn@temp @rev}%
977       {%
978         \expandafter
979         \svn@cleanfilelist\csname @svng@\svn@temp @files\endcsname
980         \svn@writegroup{\svn@temp}%
981         \@ifundefined{svng@\svn@temp @files}{}%
982         {\expandafter\svn@writeallgroups
983          \csname @svng@\svn@temp @files\endcsname
984          }%
985        }{}%
986      }%
987   \fi
988 \endgroup
989 }

```

### **\svn@updategroup**

#1: group name

Updates group with \svnfile... macro values.

```

990 \def\svn@updategroup#1{%
991   \@ifundefined{svng@#1@rev}%
992   {\svn@initgroup{#1}}%
993   {%
994     \expandafter
995     \ifnum\csname @svng@#1@rev\endcsname<\svnfilerev
996       \svn@gkwset{#1}{rev}{\svnfilerev}%
997       \svn@gkwset{#1}{date}{\svnfiledate}%
998       \svn@gkwset{#1}{author}{\svnfileauthor}%
999       \svn@gkwset{#1}{year}{\svnfileyear}%
1000      \svn@gkwset{#1}{month}{\svnfilemonth}%
1001      \svn@gkwset{#1}{day}{\svnfileday}%
1002      \svn@gkwset{#1}{hour}{\svnfilehour}%
1003      \svn@gkwset{#1}{minute}{\svnfileminute}%
1004      \svn@gkwset{#1}{second}{\svnfilesecond}%
1005      \svn@gkwset{#1}{timezonehour}{\svnfiletimezonehour}%
1006      \svn@gkwset{#1}{timezoneminute}{\svnfiletimezoneminute}%

```

```

1007     \svn@gkwset{#1}{url}{\svnfileurl}%
1008     \svn@gkwset{#1}{fname}{\svnfilefname}%
1009     \fi
1010 }

```

### **\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.

```

1011 \def\svn@definegroup#1{%
1012     \svn@gkwdef{#1}{rev}%
1013     \svn@gkwdef{#1}{date}%
1014     \svn@gkwdef{#1}{author}%
1015     \svn@gkwdef{#1}{year}%
1016     \svn@gkwdef{#1}{month}%
1017     \svn@gkwdef{#1}{day}%
1018     \svn@gkwdef{#1}{hour}%
1019     \svn@gkwdef{#1}{minute}%
1020     \svn@gkwdef{#1}{second}%
1021     \svn@gkwdef{#1}{timezonehour}%
1022     \svn@gkwdef{#1}{timezoneminute}%
1023     \svn@gkwdef{#1}{url}%
1024     \svn@gkwdef{#1}{fname}%
1025 }

```

### **\svn@initgroup**

#1: group name

Initialises group.

```

1026 \def\svn@initgroup#1{%
1027     \svn@gkwset{#1}{rev}{-1}%
1028     \svn@gkwset{#1}{date}{}%
1029     \svn@gkwset{#1}{author}{}%
1030     \svn@gkwset{#1}{year}{0000}%
1031     \svn@gkwset{#1}{month}{00}%
1032     \svn@gkwset{#1}{day}{00}%
1033     \svn@gkwset{#1}{hour}{00}%
1034     \svn@gkwset{#1}{minute}{00}%
1035     \svn@gkwset{#1}{second}{00}%
1036     \svn@gkwset{#1}{timezonehour}{+00}%
1037     \svn@gkwset{#1}{timezoneminute}{00}%
1038     \svn@gkwset{#1}{url}{}%
1039     \svn@gkwset{#1}{fname}{}%
1040 }

```

### **\svn@gkwset**

#1: group name

#2: keyword name

#3: value

Sets  $\langle value \rangle$  for  $\langle keyword \rangle$  in  $\langle group \rangle$ .

```
1041 \def\svn@gkwset#1#2#3{%
1042   \expandafter
1043   \xdef\csname @svng@#1@#2\endcsname{#3}%
1044 }
```

### **\svn@gkwdef**

#1: group name

#2: keyword name

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

```
1045 \def\svn@gkwdef#1#2{%
1046   \expandafter
1047   \xdef\csname svng@#1@#2\endcsname{\csname @svng@#1@#2\endcsname}%
1048 }
```

### **\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.

```
1049 \def\svn@cleanfilelist#1{
1050   \begingroup
1051   \def\svn@tmplist{}%
1052   \ifx\relax#1\relax\else
1053     \@for\svn@temp:=#1\do{%
1054       \expandafter\svn@ifvalidrev
1055       \expandafter{\svng@\svn@temp @rev}%
1056       {\edef\svn@tmplist{\svn@tmplist,\svn@temp}}}%
1057     }%
1058   }%
1059   \xdef#1{\expandafter\@gobble\svn@tmplist\empty}%
1060   \fi
1061 \endgroup
1062 }
1063 \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 `\ifsvnmulti@subgroups` because `\newif` inside `\if` is not a good idea.

```

1064 \newif\ifsvnsubgroups
1065 \svnsubgroupsfalse

1066 \if@svnmulti@subgroups
1067 \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.

```

1068 \def\svnsubgroup{%
1069   \begingroup
1070   \svn@removedotslash\svn@filedir
1071   \svn@subgroup{\svn@filedir\svn@filebase}%
1072   \svn@subgroup{\svn@filepath}%
1073   \endgroup
1074 }

```

### **\svn@subgroup**

**#1**: file name

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

```

1075 \def\svn@subgroup#1{%
1076   \ifnum\svnfilerev>-1\relax
1077   \expandafter\ifx\csname svn@g@#1\endcsname\relax%
1078   \expandafter\gdef\csname svn@g@#1\endcsname{1}%
1079   \svn@updategroup{#1}%
1080   \fi
1081   \fi
1082 }

```

### **\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.

```

1083 \def\svnignoreextensions#1{%
1084   \@for\svn@temp:=#1\do{%
1085     \expandafter\def\csname svn@ignore@ext@\svn@temp\endcsname{}%
1086   }%
1087 }

```

### **\svnconsiderextensions**

**#1**: A comma separated list of file name extensions (without leading dot) to consider



for automatic `\svnsubgroup`.

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

```
1088 \def\svnconsiderextensions#1{%
1089   \@for\svn@temp:=#1\do{%
1090     \expandafter\let\csname svn@ignore@ext@\svn@temp\endcsname\relax%
1091   }%
1092 }
```

The following extensions are ignored by default.

```
1093 \svnignoreextensions{aux,bbl,fd,enc,fls,glo,idx,ilg,ind,ist,%
1094 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.

```
1095 \svnmulti@atendinginputfile{%
1096   \if@svnmulti@subgroups
1097     \ifsvnsubgroups
1098       \expandafter\ifx\csname svn@ignore@ext@\svn@fileext\endcsname\relax
1099       \svnsubgroup
1100     \fi
1101   \fi
1102 \fi
1103 }
```

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.

```
1104 \if@svnmulti@subgroups
1105   \ifsvnsubgroups
1106     \svn@addfiletogroup{\jobname .\svn@mainext}{\jobname}%
1107     \svnsubgroup
1108   \fi
1109 \fi
1110 \AtBeginDocument{%
1111   \if@svnmulti@subgroups
1112     \ifsvnsubgroups
1113       \@ifundefined{@svng@\svn@g @files}%
1114       {%
1115         \expandafter
1116         \xdef\csname @svng@\svn@g @files\endcsname{\jobname}%
1117       }%
1118       {%
1119         \expandafter
1120         \xdef\csname @svng@\svn@g @files\endcsname
1121         {\jobname,\csname @svng@\svn@g @files\endcsname}%
1122       }%
1123       \svnsubgroup
1124     \fi
1125   \fi
```

```

1126 }
1127 \fi

```

## 7.16 External Files

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

```

1128 \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.

```

1129 \if@svnmulti@groups
1130 \def\svnexternalgroup#1{%
1131   \svn@ifequal{#1}{*}%
1132     {\def\svn@externalgroup{#1}}%
1133     {\def\svn@externalgroup{\svn@g}}%
1134 }
1135 \def\svn@externalgroup{\svn@g}
1136 \else
1137 \def\svn@externalgroup{}
1138 \fi

```

### `\svnexternal`

**#1:** group name

**#2:** list of filenames in { }

```

1139 \if@svnmulti@autokw
1140 \newcommand*\svnexternal[2][]{%
1141   \svn@pushfilestack
1142   \svn@ifequal{#1}{*}%
1143     {\edef\svn@eg{\svn@g}}%
1144     {\svn@ifempty{#1}%
1145       {\edef\svn@eg{\svn@externalgroup}}%
1146       {\edef\svn@eg{#1}}%
1147     }%
1148   \svne@@external#2\relax
1149   \svn@popfilestack
1150 }
1151
1152 \def\svne@@external#1{%
1153   \ifx\relax#1\empty\else
1154     \svnegetfile{#1}%
1155     \begingroup\svn@externalfile{\svn@eg}{#1}%
1156     \expandafter\svne@@external
1157   \fi

```

```

1158 }
1159 \else
    Writes the current input file path and its argument as arguments of \svnexternal
    into the .svn file.
1160 \newcommand*{\svnexternal[2][]{%
1161     \if@filesw
1162         \svn@checkwrite
1163         \begingroup
1164             \svn@ifequal{#1}{*}%
1165             {\def\svn@temp{\svn@g}}%
1166             {\svn@ifempty{#1}%
1167              {\def\svn@temp{\svn@externalgroup}}%
1168              {\def\svn@temp{#1}}%
1169             }%
1170             \immediate\write\svn@write{%
1171               \noexpand\svnexternal[\svn@temp]{\svn@filepath}{#2}%
1172             }%
1173         \endgroup
1174     \fi
1175     \svn@inputsvx{\svn@filedir\svn@filebase}%
1176 }
1177 \fi

```

#### \svnexternalpath

#1: list of paths in { }

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

```

1178 \def\svnexternalpath#1{%
1179     \if@filesw
1180         \svn@checkwrite
1181         \immediate\write\svn@write{%
1182           \noexpand\svnexternalpath{#1}%
1183         }%
1184     \fi
1185 }

```

#### \@svnexternal

#### \@svnexternalpath

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

```

1186 \newcommand*\@svnexternal[3][]{}
1187 \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.

```

1188 \newcommand*\svnexternalfile[1][\svn@filedir\svn@filebase]{%
1189   \begingroup % TODO: maybe use \svn@catcodes
1190     \catcode'\_ =12
1191     \catcode'\& =12
1192     \catcode'\^ =12
1193     \catcode'\$ =12
1194     \catcode'\# =12
1195     \svn@externalfile{#1}%
1196 }
```

### `\svn@externalfile`

#1: group name

#2: file name

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

```

1197 \def\svn@externalfile#1#2{%
1198   \endgroup
1199   \if@svnmulti@subgroups
1200     \ifsvnsubgroups
1201       \svn@ifequal{#1}{\svn@rg}%
1202       {\svn@addfiletogroup{#2}{\svn@filedir\svn@filebase}}%
1203       {\svn@addfiletogroup{#2}{#1}}%
1204       \svn@subgroup{#2}%
1205     \fi
1206   \fi
1207 }
```

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

```

1208 \else
1209   \def\svnexternalfile#1{%
1210     \fi
```

## 7.17 Auto loading of .svx files

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

```

1211 \if@svnmulti@autoload
1212
1213 \svnmulti@atbegininputfile{%
1214   \svn@ifequal{\svn@fileext}{tex}%
1215   {\svn@inputsvx{\svn@filedir\svn@filebase}}%
1216   {}%
1217 }
```

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

```

1218 %%\AtEndOfPackage{%
```

```

1219 \AtBeginDocument{%
1220   \svn@inputsvx{\jobname}%
1221 }
1222 \fi

```

## 7.18 Support for Graphic Packages

### 7.18.1 Common Code

```

1223 \if@svnmulti@anygraphic

```

#### `\svngraphicsgroup`

#1: graphic group name

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

```

1224 \def\svngraphicsgroup#1{%
1225   \svn@ifequal{#1}{*}%
1226   {\def\svn@graphicsgroup{\svn@g}}%
1227   {\def\svn@graphicsgroup{#1}}%
1228 }
1229 \def\svn@graphicsgroup{\svn@externalgroup}

```

#### `\svnignoregraphic`

#1: file name/path

Ignores the given graphic file by defining a special macro.

```

1230 \def\svnignoregraphic#1{%
1231   \expandafter\def\csname svn@ignoregraphic@#1\endcsname{%
1232 }

```

#### `\svnconsidergraphic`

#1: file name/path

Deletes the special ignore macro to consider the graphic again.

```

1233 \def\svnconsidergraphic#1{%
1234   \expandafter\let\csname svn@ignoregraphic@#1\endcsname\relax%
1235 }
1236 \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.

```

1237 \if@svnmulti@graphics
1238 \RequirePackage{graphics}[2006/02/20]

```

#### `\Gin@setfile`

#1: ??, not used

```

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

1239 \message{Package svn-multi: patching macro '\string\Gin@setfile' from the
1240 'graphics' package!}%
1241 \let\svnmulti@Gin@setfile\Gin@setfile
1242 \renewcommand*{\Gin@setfile}[3]{%
1243   \expandafter\ifx\csname svn@ignoregraphic@#3\endcsname\relax%
1244     \svnexternal[\svn@graphicsgroup]{#3}%
1245   \fi
1246   \svnmulti@Gin@setfile{#1}{#2}{#3}%
1247 }
1248 \fi

```

### 7.18.3 Package pgf

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

```

1249 \if@svnmulti@pgfimages
1250 \RequirePackage{pgf}[2008/01/15]

```

#### `\pgf@declareimage`

```

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

```

```

1251 \message{Package svn-multi: patching macro '\string\pgf@declareimage' and will
1252 patch generated macros '\string\pgf@image@<name>!' from the 'pgf' package!}%
1253 \let\svnmulti@pgf@declareimage\pgf@declareimage
1254 \renewcommand*{\pgf@declareimage}[3][]{%
1255   \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`.

```

1256   \ifx\pgf@filename\empty\else
1257     \expandafter\ifx\csname svn@ignoregraphic@\pgf@filename\endcsname\relax%
1258       \expandafter\global\expandafter%
1259       \let\csname svnmulti@pgf@image@#2!\endcsname=\pgf@image%
1260       \expandafter\xdef\csname pgf@image@#2!\endcsname{%
1261         \noexpand\svnexternal[\noexpand\svn@graphicsgroup]{\pgf@filename}}%
1262       \csname svnmulti@pgf@image@#2!\endcsname
1263     }%
1264   \fi
1265 \fi
1266 }
1267 \fi

```

## 7.19 Table of Revisions

```
1268 \if@svnmulti@table
1269 \ifx\tableofcontents\relax\else
```

### **\svnrevisionsname**

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

```
1270 \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.

```
1271 \def\svn@svt{svt}
```

### **\tableofrevisions**

The \tableofcontents macro from standard L<sup>A</sup>T<sub>E</sub>X is adapted for this macro. Classes which provide chapters will get a different table then one which not.

```
1272 \AtBeginDocument{%
1273 \ifx\chapter\@undefined
1274
1275 %% Adapted from the \tableofcontents macro, LaTeX ‘article’ class [2005/09/16]
1276 \newcommand\tableofrevisions{%
1277   \section*{\svnrevisionsname
1278     \@mkboth{\MakeUppercase\svnrevisionsname}{\MakeUppercase\svnrevisionsname}}%
1279   \svn@input{\jobname .\svn@svt}%
1280 }
1281
1282 \else
1283
1284 %% Adapted from the \tableofcontents macro, LaTeX ‘book’ class [2005/09/16]
1285 \newcommand\tableofrevisions{%
1286   \if@twocolumn
1287     \@restonecoltrue\onecolumn
1288   \else
1289     \@restonecolfalse
1290   \fi
1291   \chapter*{\svnrevisionsname
1292     \@mkboth{\MakeUppercase\svnrevisionsname}{\MakeUppercase\svnrevisionsname}}%
1293   \svn@input{\jobname .\svn@svt}%
1294   \if@restonecol\twocolumn\fi
1295 }
1296
1297 \fi
1298 }
1299 \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.

```
1300 \def\svn@writerow#1#2#3{%
1301   \immediate\write\svn@svtwrite{%
1302     \expandafter\noexpand\csname svn#1row\endcsname
1303     \expandafter\noexpand\csname svntab#1\endcsname{#2}{#3}\space
1304     \@ampersamchar\space
1305     \svn@tabcell{rev}\space\@ampersamchar\space
1306     \svn@tabcell{author}\space\@ampersamchar\space
1307     \noexpand\svntabdate%
1308     \svn@tabcellarg{year}%
1309     \svn@tabcellarg{month}%
1310     \svn@tabcellarg{day}%
1311     \svn@tabcellarg{hour}%
1312     \svn@tabcellarg{minute}%
1313     \svn@tabcellarg{second}%
1314     \svn@tabcellarg{timezonehour}%
1315     \svn@tabcellarg{timezoneminute}%
1316     \space\@backslashchar\@backslashchar
1317     \expandafter\noexpand\csname endsvn#1row\endcsname
1318   }%
1319 }
```

#### `\svn@writegrouprow`

#1: current group

```
1320 \def\svn@writegrouprow#1{%
1321   \begingroup
1322     \def\svn@tabcellarg##1{{\csname @svng@#1##1\endcsname}}%
1323     \def\svn@tabcell##1{\expandafter\noexpand\csname svntab##1\endcsname%
1324       \svn@tabcellarg{##1}%
1325     }%
1326     \svn@writerow{group}{#1}{}%
1327   \endgroup
1328 }
```

#### `\svn@writesubgrouprow`

#1: grouping level

#2: subgroup name

```
1329 \def\svn@writesubgrouprow#1#2{%
1330   \begingroup
1331     \def\svn@tabcellarg##1{{\csname @svng@#2@##1\endcsname}}%
1332     \def\svn@tabcell##1{\expandafter\noexpand\csname svntab##1\endcsname%
1333       \svn@tabcellarg{##1}%
1334     }%
1335     \svn@writerow{subgroup}{#1}{#2}%
1336   \endgroup
```



```
1337 }
```

#### **\svn@writefilerow**

#1: grouping level

#2: file name

```
1338 \def\svn@writefilerow#1#2{%
1339   \begingroup
1340   \def\svn@tabcellarg##1{{\csname @svng@#2@##1\endcsname}}%
1341   \def\svn@tabcell##1{\expandafter\noexpand\csname svntab##1\endcsname%
1342     \svn@tabcellarg{##1}%
1343   }%
1344   \svn@writerow{file}{#1}{#2}%
1345   \endgroup
1346 }
```

#### **\svn@writeglobalrow**

```
1347 \def\svn@writeglobalrow{%
1348   \begingroup
1349   \def\svn@tabcellarg##1{{\csname @svn@##1\endcsname}}%
1350   \def\svn@tabcell##1{\expandafter\noexpand\csname svntab##1\endcsname%
1351     \svn@tabcellarg{##1}%
1352   }%
1353   \svn@writerow{global}{-}{-}%
1354   \endgroup
1355 }
```

### **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**

```
1356 \def\svntable{%
1357   \begin{tabular}{p{0.425\textwidth}rll}%
1358   \hline
1359 }
```

#### **\endsvntable**

```
1360 \def\endsvntable{\hline\end{tabular}}
```

#### **\svntablehead**

```
1361 \def\svntablehead{%
1362   Name & Rev & Last Author & Last Changed At \\\hline
1363 }
```

#### **\svntablefoot**

```
1364 \def\svntablefoot{}
```

#### **\svnbeforetable**

```
1365 \def\svnbeforetable{}
```

```

\svnaftertable
1366 \def\svnaftertable{\clearpage}

\svnglobalrow
1367 \def\svnglobalrow{}

\endsvnglobalrow
1368 \def\endsvnglobalrow{}

\svngrouprow
1369 \def\svngrouprow{}

\endsvngrouprow
1370 \def\endsvngrouprow{}

\svnsubgrouprow
1371 \def\svnsubgrouprow{}

\endsvnsubgrouprow
1372 \def\endsvnsubgrouprow{}

\svnsubgrouprow
1373 \def\svnfilerow{}

\endsvnfilerow
1374 \def\endsvnfilerow{}

\svntabglobal
1375 \def\svntabglobal{Document}

\svntabgroup
1376 \def\svntabgroup#1{Group '#1'}

\svntabfile
1377 \def\svntabsubgroup#1{%
1378   \raggedright
1379   \addtolength{\leftskip}{#1\medskipamount}%
1380   \begingroup
1381   \catcode'\_ =12
1382   \catcode'\& =12
1383   \catcode'\^ =12
1384   \catcode'\$ =12
1385   \catcode'\# =12
1386   \svn@tabsubgroup
1387 }
1388 \def\svn@tabsubgroup#1{\endgroup Subgroup '\texttt{\small #1}'}

```

### `\svntabfile`

```
1389 \def\svntabfile#1{%
1390   \raggedright
1391   \addtolength{\leftskip}{#1\medskipamount}%
1392   \begingroup
1393   \catcode'\_ =12
1394   \catcode'\&=12
1395   \catcode'\^ =12
1396   \catcode'\$=12
1397   \catcode'\#=12
1398   \svn@tabfile
1399 }
1400 \def\svn@tabfile#1{\endgroup File ‘\texttt{\small #1}’}
```

### `\svntabrev`

```
1401 \def\svntabrev{}
```

### `\svntabauthor`

```
1402 \def\svntabauthor#1{\svnFullAuthor{#1}}
```

### `\svntabdate`

```
1403 \def\svntabdate#1#2#3#4#5#6#7#8{%
1404   #1-#2-#3 #4:#5:#6% #7#8%
1405 }
1406 \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.

```
1407 \newcommand{\svn}{\@ifnextchar{*}{\svn@s}{\svn@n}}
1408 \def\svn@n#1{\@svn@n#1}
1409 \def\svn@s*#1{\@svn@s#1}
1410 \def\@svn@n$#1$#1{#1}
1411 \def\@svn@s$#1 $#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.

```

1412 %% Adapted from the \url macro of the ‘hyperref’ package.
1413 \DeclareRobustCommand*\svnnolinkurl{%
1414   \@ifundefined{hyper@normalise}%
1415     {\PackageWarning{svn-multi}{Package hyperref is needed for \noexpand
1416       \svnnolinkurl.}}%
1417     {\hyper@normalise\svnnolinkurl@}%
1418 }%
1419 \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.

```

1420 \def\svn@getfilename#1{%
1421   \begingroup
1422     \gdef\svnfiledir{%
1423       \edef\svn@temp{#1}%
1424       \expandafter\@svn@getfilename\svn@temp/{ }\relax
1425 }%

```

### **\@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.

```

1426 \def\@svn@getfilename#1/#2\relax{%
1427   \svn@ifempty{#2}%
1428     {%
1429       \endgroup
1430       \gdef\svnfilefname{#1}%
1431     }%
1432     {%
1433       \xdef\svnfiledir{\svnfiledir#1/}%
1434       \@svn@getfilename#2\relax
1435     }%
1436 }%

```

## **7.21 Auxiliary file generation and read-back**

Reread output from last compile run if it exists.

```

1437 \@input{\jobname .svn}

```

### **\svn@checkwrite**

Checks if `.svn` file is already open and if not opens it. This makes sure that the file

is only created if really needed. The macro is only needed once, so it's redefines itself to `\relax` at the end.

```

1438 \def\svn@checkwrite{%
1439   \@ifundefined{svn@write}{%
1440     \newwrite\svn@write
1441     \immediate\openout\svn@write=\jobname.svn\relax%
1442     \immediate\write\svn@write{\@percentchar\space SVN Keyword cache}%
1443     %\immediate\write\svn@write{\noexpand\makeatletter}%
1444   }{}%
1445   \let\svn@checkwrite=\relax
1446 }

```

### `\svn@writesvn`

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

```

1447 {\catcode'\&=12
1448 \gdef\@ampersamchar{&}
1449 }
1450 \def\svn@writesvn{%

```

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

```

1451   \if@svnmulti@groups
1452   \fi

```

Write document global values:

```

1453   \svn@checkwrite
1454   \immediate\write\svn@write{^^J%
1455     \@percentchar\space Global values:^^J%
1456     \noexpand\def\noexpand\svnrev{\@svn@rev}^^J%
1457     \noexpand\def\noexpand\svndate{\@svn@date}^^J%
1458     \noexpand\def\noexpand\svnauthor{\@svn@author}^^J%
1459     \noexpand\def\noexpand\svnyear{\@svn@year}^^J%
1460     \noexpand\def\noexpand\svnmonth{\@svn@month}^^J%
1461     \noexpand\def\noexpand\svnday{\@svn@day}^^J%
1462     \noexpand\def\noexpand\svnhour{\@svn@hour}^^J%
1463     \noexpand\def\noexpand\svnminute{\@svn@minute}^^J%
1464     \noexpand\def\noexpand\svnsecond{\@svn@second}^^J%
1465     \noexpand\def\noexpand\svntimezonehour{\@svn@timezonehour}^^J%
1466     \noexpand\def\noexpand\svntimezoneminute{\@svn@timezoneminute}^^J%
1467     \noexpand\svn@gdefverb\noexpand\svnurl{\@svn@url}^^J%
1468     \noexpand\svn@gdefverb\noexpand\svnfname{\@svn@fname}^^J%
1469   }%

```

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

```

1470   \if@svnmulti@groups
1471     \svn@cleanfilelist\@svng@@files
1472     \immediate\write\svn@write{%
1473       \noexpand\def\noexpand\svng@@files{\@svng@@files}^^J%
1474     }%
1475     \svn@writeallgroups\@svng@@files

```

Write keyword group values if groups were specified:

```

1476     \ifx\svn@glst\empty\else
1477     \begingroup
1478     \immediate\write\svn@write{^^J%
1479     \@percentchar\space SVN File Groups: \svn@glst
1480     }%
1481     \svn@writeallgroups\svn@glst
1482     \endgroup
1483     \fi
1484     \else
1485     \immediate\write\svn@write{^^J}%
1486     \fi

```

Finally close output file:

```

1487     \immediate\closeout\svn@write%
1488 }

```

### **\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.

```

1489 \def\svn@writegroupfiles#1{%
1490   \begingroup
1491   \advance\svn@grouplevel by 1\relax
1492   \expandafter\let
1493   \expandafter\svn@files\csname @svng@#1@files\endcsname

```

Stop if no files in list.

```

1494   \ifx\svn@files\relax\else
1495   \svn@cleanfilelist\svn@files
1496   \@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.

```

1497     \svn@ifvalidrev{@svng@\svn@file @rev}%
1498     {%
1499     \@ifundefined{@svng@\svn@file @files}%
1500     {\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.

```

1501     {\svn@ifonlyone{\svn@file}%
1502     {\svn@writefilerow{\the\svn@grouplevel}%
1503     {\csname @svng@\svn@file @files\endcsname}}}%
1504     {\svn@ifempty{\csname @svng@\svn@file @files\endcsname}%
1505     {}%
1506     {%
1507     \svn@writesubgrouprow{\the\svn@grouplevel}{\svn@file}%
1508     \svn@writegroupfiles{\svn@file}%

```

```

1509         }%
1510     }%
1511 }%
1512 }{}%
1513 }%
1514 \fi
1515 \endgroup
1516 }%

```

### **\svn@writesvt**

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

```

1517 \def\svn@writesvt{%
    Write table of revisions if enabled.
1518 \if@svnmulti@table
    Open .svt file and write table header:
1519 \newwrite\svn@svtwrite
1520 \immediate\openout\svn@svtwrite=\jobname.\svn@svt\relax
1521 \@onelevel@sanitize\svntable%
1522 \immediate\write\svn@svtwrite{%
1523     \noexpand\svnbeforetable^^J%
1524     \svntable
1525     \noexpand\svntablehead^^J%
1526 }%
    Group rows:
1527 \let\svn@grouplevel\@tempcnta
1528 \svn@grouplevel=0\relax
1529 \svn@writeglobalrow{}%
1530 \svn@writegroupfiles{}%
1531 %
1532 \@for\svn@g:=\svn@glist\do{%
1533     \@ifundefined{svng@\svn@g @rev}{}%
1534     {%
1535         \expandafter
1536         \ifnum\csname @svng@\svn@g @rev\endcsname>0\relax
1537             \svn@writegrouprow{\svn@g}%
1538             \svn@writegroupfiles{\svn@g}%
1539         \fi
1540     }%
1541 }%
1542
    Write table footer and close file:
1543 \@onelevel@sanitize\endsvntable%
1544 \immediate\write\svn@svtwrite{%
1545     \noexpand\svntablefoot^^J%
1546     \endsvntable^^J%
1547     \noexpand\svnaftertable

```

```

1548 }%
1549 \immediate\closeout\svn@svtwrite%
1550 \fi
1551 }

```

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.

```

1552 \if@svnmulti@autokwall
1553
1554 \svnmulti@atbegininputfile{%
1555   \expandafter
1556   \ifx\csname svn@ignore@ext@\svn@fileext\endcsname\relax
1557     \svnnegetfile{\svn@filepath}%
1558   \fi
1559 }
1560
1561 \fi

```

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

```

1562 \AtEndDocument{%
1563   \if@filesw
1564     \ifx\@svn@rev\empty\else
1565       \ifnum\@svn@rev<1\else
1566         \svn@writesvn
1567         \svn@writesvt
1568       \fi
1569     \fi
1570   \fi
1571 }

```

## 7.22 Backward compatibility wrapper `svnk.w.sty`

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

```

1 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
2 \ProvidesPackage{svnk.w}
3   [2009/03/27 v2.1 Backward compatibility wrapper for svn-multi]
4 \PackageWarning{svnk.w}{The package 'svnk.w' got renamed to 'svn-multi' and is now
5 only a backward compatibility wrapper which loads 'svn-multi'. Please adjust
6 your document preamble to use the new name.}
7 \RequirePackage{svn-multi}

```



## Change History

v1.0	General: Initial version . . . . .	1	spaces generated by <code>\svnid</code> , <code>\svnidlong</code> and <code>\svnkwsave</code> , e.g. when used in a file which is included with <code>\input</code> . . . . .	1
v1.1	General: Added macros to extract and typeset date/time informa- tion. Added macros to set and typeset main URL or filename. .	1		
v1.2	General: Added star version. Nor- mal version was not changed to not break existing docu- ments with user defined key- words without leading space. .	67	v1.3b General: Changed the way catcodes are modified to be compatible with the french option of the ba- bel package or other packages which modify the list of special characters. . . . .	1
	Added warning when a wrong, maybe misspelled, keyword is given. . . . .	40	v1.4 General: Added support for time- zones with non-zero minute part, e.g. +0530. . . . .	1
	Macro now returns the username if the full name was not regis- tered. . . . .	44	Return now full timezone (hour + minute part). Manually added 00 minutes are removed. .	32
	New macro . . . . .	43–45, 67	v1.5 General: Added <code>\today</code> -style macros <code>\svntoday</code> and <code>\svnfiletoday</code> . . . . .	1
	Renamed package from <code>svnkw</code> to <code>svn-multi</code> to match CTAN directory. Wrapper file <code>svnkw.sty</code> is provided for back- ward compatibility. . . . .	1	v2.0 General: New features: keyword groups, external file support, auto-include of images, files as subgroups and table of revi- sions. . . . .	1
v1.3	General: Added verbatim support. Keywords can now contain spe- cial character like <code>_ ^ \$ % &amp; \</code> . Rewrote keyword check macros to work with verbatim code. <code>\nofiles</code> is now obeyed. . . . .	1	v2.1 General: Added automatic extrac- tion of keywords from Subver- sion 'entries' file: 'autokw' op- tion . . . . .	1
v1.3a	General: Fixed issue with unwanted			

## Index

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

<b>Symbols</b>	1193, 1384, 1396	<code>\.</code> . . . . .	252, 317
<code>\#</code> . . . .	1194, 1385, 1397	<code>\&amp;</code> 1191, 1382, 1394, 1447	<code>\:</code> . . . . . 252, 316
<code>\\$</code> . . . . .	555, 562,	<code>\-</code> . . . . .	252, 315
		<code>\@@input</code> . . . . .	195



\if@svnmulti@external	S	\svn@fileext .. 756,
..... 60, 1128	\svn ..... 1407	758, 772, 785,
\if@svnmulti@filehooks	\svn@ifonlyone 186, 189	934, 936, 938,
..... 726	\svn@addfiletogroup	1098, 1214, 1556
\if@svnmulti@graphics	.... 917, 935,	\svn@filename . 758, 759
.... 67, 128, 1237	937, 939, 940,	\svn@filepath . 759,
\if@svnmulti@groups	1106, 1202, 1203	767, 931, 940,
..... 57,	\svn@catcodes .....	1072, 1171, 1557
470, 766, 871,	. 494, 503, 512,	\svn@files ... 1493,
1129, 1451, 1470	538, 628, 644, 1189	1494, 1495, 1496
\if@svnmulti@pgfimages	\svn@cgs 220, 426, 895,	\svn@filestack 799,
.... 72, 131, 1249	896, 901, 902, 911	802, 818, 827,
\if@svnmulti@subgroups	\svn@checkcolon 558, 602	828, 853, 856, 858
..... 63,	\svn@checkwrite ...	\svn@finkerror ....
474, 1066, 1096,	... 1162, 1180,	.... 729, 733,
1104, 1111, 1199	1438, 1445, 1453	735, 736, 737,
\if@svnmulti@table .	\svn@cleanfilelist .	738, 739, 740, 741
... 81, 1268, 1518	..... 979,	\svn@g .... 194, 196,
\if@svnmulti@verbatim	1049, 1471, 1495	220, 471, 472,
.... 493, 502,	\svn@definegroup . 1011	484, 768, 816,
514, 554, 627, 643	\svn@depooption 50, 58,	848, 878, 879,
\ifeof ..... 178,	61, 64, 65, 68,	880, 882, 883,
255, 262, 281, 286	69, 70, 73, 74,	892, 895, 1113,
\IfFileExists ..... 195	75, 78, 79, 82,	1116, 1120,
\ifsvnsubgroups ...	83, 92, 93, 110, 116	1121, 1133,
475, 1064, 1097,	\svn@eg ..... 1143,	1135, 1143,
1105, 1112, 1200	1145, 1146, 1155	1165, 1226,
	\svn@externalfile .	1532, 1533,
M	.. 1155, 1195, 1197	1536, 1537, 1538
\message ..... 52,	\svn@externalgroup .	\svn@gdefverb .....
761, 778, 1239, 1251	... 1132, 1133,	.. 623, 1467, 1468
	1135, 1137,	\svn@gdefverb@ ....
N	1145, 1167, 1229	.... 630, 632, 646
\newread ..... 247	\svn@file 1496, 1497,	\svn@getfilename ..
\next . 263, 267, 269,	1499, 1500,	... 383, 452, 1420
272, 279, 290,	1501, 1503,	\svn@gkwdef ... 1012,
298, 746, 747, 748	1504, 1507, 1508	1013, 1014,
	\svn@filebase .....	1015, 1016,
O	. 476, 755, 758,	1017, 1018,
\openin ..... 280, 285	769, 771, 784,	1019, 1020,
	935, 937, 939,	1021, 1022,
P	940, 1071, 1175,	1023, 1024, 1045
\pgf@declareimage .	1188, 1202, 1215	\svn@gkwset ... 996,
.. 1251, 1253, 1254	\svn@filedir .. 476,	997, 998, 999,
\pgf@filename .....	757, 759, 769,	1000, 1001,
.. 1256, 1257, 1261	773, 774, 786,	1002, 1003,
\pgf@image ..... 1259	787, 935, 937,	1004, 1005,
\pgf@image@ ..... 1252	939, 940, 1070,	1006, 1007,
	1071, 1175,	1008, 1027,
R	1188, 1202, 1215	1028, 1029,
\read ..... 258, 265		

1030, 1031,	\svn@nextchar . 400, 405	\svn@temp ... 89, 90,
1032, 1033,	\svn@pg ... 220, 768,	95, 96, 106, 107,
1034, 1035,	769, 817, 849,	112, 113, 118,
1036, 1037,	879, 935, 937, 939	119, 123, 124,
1038, 1039, 1041	\svn@popfilestack .	136, 137, 164,
\svn@glst 872, 884,	202, 852, 867, 1149	166, 167, 218,
885, 887, 1476,	\svn@pushfilestack .	265, 266, 307,
1479, 1481, 1532	200, 801, 863, 1141	350, 351, 352,
\svn@gobblezeros ..	\svn@removedotslash	353, 354, 355,
.... 396, 397, 399	..... 743,	625, 634, 640,
\svn@gobblezeros@ .	754, 774, 787, 1070	693, 694, 697,
..... 400, 402	\svn@removedotslash@	698, 714, 715,
\svn@graphicsgroup .	.... 744, 746, 750	718, 719, 913,
... 1226, 1227,	\svn@restorefilekws	915, 975, 976,
1229, 1244, 1261	..... 821, 858	979, 980, 981,
\svn@grouplevel 1491,	\svn@restorefilekws@	983, 1053, 1055,
1500, 1502,	..... 822, 830	1056, 1084,
1507, 1527, 1528	\svn@restorefilekws@@	1085, 1089,
\svn@ifempty 134, 190,	..... 840, 843	1090, 1165,
304, 576, 603,	\svn@rg .....	1167, 1168,
608, 826, 1144,	194, 196, 220, 1201	1171, 1423, 1424
1166, 1427, 1504	\svn@s .....	\svn@tmplist .....
\svn@ifeof 177, 339, 344	\svn@scanId ... 439, 547	.. 1051, 1056, 1059
\svn@ifequal .. 147,	\svn@stringa .. 149, 151	\svn@updategroup ..
305, 348, 370,	\svn@stringb .. 150, 151	472, 476, 990, 1079
610, 616, 745,	\svn@stripcolon 605, 607	\svn@value ... 605,
767, 856, 874,	\svn@stripspace 612, 615	609, 611, 617, 621
894, 931, 934,	\svn@striptrailingspace	\svn@write 944, 960,
936, 938, 1131,	..... 618, 620	965, 1170, 1181,
1142, 1164,	\svn@subgroup . 1071,	1440, 1441,
1201, 1214, 1225	1072, 1075, 1204	1442, 1443,
\svn@ifonlyone 184, 1501	\svn@svt .....	1454, 1472,
\svn@ifvalidrev ...	1271, 1279, 1293, 1520	1478, 1485, 1487
161, 976, 1054, 1497	\svn@svtwrite . 1301,	\svn@writeallgroups
\svn@initfile .....	1519, 1520,	972, 982, 1475, 1481
.... 223, 239, 933	1522, 1544, 1549	\svn@writefilerow .
\svn@initgroup .....	\svn@tabcell .. 1305,	.. 1338, 1500, 1502
... 883, 992, 1026	1306, 1323,	\svn@writeglobalrow
\svn@input .....	1332, 1341, 1350	..... 1347, 1529
192, 201, 1279, 1293	\svn@tabcellarg 1308,	\svn@writegroup 942, 980
\svn@inputsvx . 199,	1309, 1310,	\svn@writegroupfiles
1175, 1215, 1220	1311, 1312,	..... 1489,
\svn@maindir .....	1313, 1314,	1508, 1530, 1538
. 753, 754, 757, 931	1315, 1322,	\svn@writegrouprow .
\svn@mainext .....	1324, 1331,	..... 1320, 1537
. 392, 752, 756,	1333, 1340,	\svn@writekw 943, 948,
767, 931, 934, 1106	1342, 1349, 1351	949, 950, 951,
\svn@n .....	\svn@tabfile 1398, 1400	952, 953, 954,
\svn@namegdefverb .	\svn@tabsubgroup ..	955, 956, 957, 958
. 637, 962, 967, 969	..... 1386, 1388	

\svn@writerow . . . . .	\svne@name . . . . .	\svnfilehour 230, 328,
. . . 1300, 1326,	. 302, 304, 342,	411, 461, 654,
1335, 1344, 1353	343, 361, 370, 372	666, 809, 837, 1002
\svn@writesubgrouprow	\svne@parsedirentory	\svnfileminute . . . .
. . . . . 1329, 1507	. . . . . 290, 301	. . . . 231, 329,
\svn@writesvn 1450, 1566	\svne@parseentries .	411, 462, 655,
\svn@writesvt 1517, 1567	. . . . 291, 338, 365	667, 810, 838, 1003
\svn@aftertable . . . .	\svne@parseentriesfile	\svnfilemonth . . . . .
. . . . . 1366, 1547	. . . . 277, 386, 387	. . . . 228, 326,
\svnauthor . . 207, 1458	\svne@read . . . 247,	421, 459, 652,
\svnbeforetable . . .	255, 258, 262,	664, 807, 835, 1000
. . . . . 1365, 1523	265, 275, 280,	\svnfilerev . . . . .
\svncg@def . . . 898, 909	281, 283, 285,	. 224, 448, 453,
\svncgday . . . . . 428	286, 293, 339, 344	454, 455, 803,
\svncghour . . . . . 412	\svne@readline . . . .	831, 995, 996, 1076
\svncgminute . . . . . 412	. . . . 254, 282,	\svnfilerow . . . . . 1373
\svncgmonth . . . . . 428	288, 302, 303,	\svnfilesecond . . . .
\svncgsecond . . . . . 412	307, 308, 342,	. . . . 232, 330,
\svncgtime . . . . . 412	347, 350, 351,	411, 463, 656,
\svncgtimezone . . . . 398	352, 353, 354,	668, 811, 839, 1004
\svncgtimezonehour . 398	355, 356, 357, 358	\svnfiletime . . . . . 411
\svncgtimezoneminute	\svne@rev . 357, 374, 377	\svnfiletimezone . . 397
. . . . . 398	\svne@scandate 319, 360	\svnfiletimezonehour
\svncgtoday . . . . . 425	\svne@scandate@ 320, 324	. . . . 233, 331,
\svncgyear . . . . . 428	\svne@url . 361, 376, 377	397, 464, 657,
\svnconsiderextensions	\svne@version . . . . .	672, 812, 844, 1005
. . . . . 1088	. 282, 284, 288, 289	\svnfiletimezoneminute
\svnconsidergraphic	\svnegetfile . . . . .	. . . . 234, 332,
. . . . . 1233	381, 392, 1154, 1557	397, 465, 658,
\svndate . . . 206, 1457	\svnexternal . . 1140,	673, 813, 845, 1006
\svnday . . . . .	1160, 1244, 1261	\svnfiletoday . . . . 419
210, 415, 676, 1461	\svnexternalfile . .	\svnfileurl . . . 235,
\svne@@external . . .	. . . . . 1188, 1209	451, 452, 466,
. . 1148, 1152, 1156	\svnexternalgroup 1130	680, 814, 846, 1007
\svne@author . . . . .	\svnexternalpath . 1178	\svnfileyear 227, 325,
. . . . 358, 375, 377	\svnfileauthor . . . .	421, 458, 651,
\svne@baseurl . 308, 361	. . . . 226, 450,	663, 806, 834, 999
\svne@catcodes 248, 287	457, 805, 833, 998	\svnfname . . . 217, 1468
\svne@date 334, 356,	\svnfiledate . . . . .	\svnFullAuthor 687, 1402
359, 360, 373, 377	. 225, 333, 449,	\svnFullAuthor@ . . .
\svne@endread . 274, 378	456, 804, 832, 997	. . . . 694, 698, 700
\svne@ff . . . . . 243, 266	\svnfileday 229, 327,	\svnFullAuthor@normal
\svne@gobblertest . .	421, 460, 653,	. . . . . 690, 696
. 261, 269, 309, 364	665, 808, 836, 1001	\svnFullAuthor@star
\svne@handleentry .	\svnfiledir . . . 237,	. . . . . 689, 692
. . . . . 362, 369	386, 387, 1422, 1433	\svnFullRevision . . 708
\svne@kind . . . . .	\svnfilename . . . . .	\svnFullRevision@ .
. 303, 305, 347, 348	. 236, 370, 384,	. . . . 715, 719, 721
	385, 467, 681,	\svnFullRevision@normal
	815, 847, 1008, 1430	. . . . . 711, 717

\svnFullRevision@star	\svnkwnkdef@URL	582	\svnnolinkurl@	....
..... 710, 713	\svnkwnkHeadURL	591	..... 1417, 1419	
\svng	\svnkwnkId	545, 547	\svnpdfdate	..... 675
\svng@@files	\svnkwnkLastChangedAt	590	\svnRegisterAuthor	. 684
\svnglobalrow	\svnkwnkLastChangedBy	589	\svnRegisterRevision	
\svnggraphicsgroup	\svnkwnkLastChangedDate	..... 705		
\svnggroup	..... 529, 530		\svnrev	..... 205, 1456
\svnggrouprow	\svnkwnkLastChangedRevision		\svnrevisionsname	.
\svnhour	..... 588		... 1270, 1277,	
211, 410, 677, 1462	\svnkwnkRev	587, 588	1278, 1291, 1292	
\svnid	\svnkwnkRevision	587	\svnsecond	.....
\svnidlong	\svnkwnksave	435, 536	213, 410, 677, 1464	
\svnidlong@readargs	\svnkwnksave@parse	563	\svnsetcg	..... 893
.... 498, 519, 523	\svnkwnksave@read	...	\svnsetmainfile	679, 683
\svnidlong@readargsfull	.... 524, 525,		\svnsubgroup	.. 1068,
.... 498, 500, 518	526, 527, 542, 557		1099, 1107, 1123	
\svnignoreextensions	\svnkwnksave@readargs		\svnsubgrouprow	.. 1371
..... 1083, 1093	..... 539, 541		\svnsubgroupsfalse	1065
\svnignoregraphic	\svnkwnkURL	591	\svnsubgroupstrue	1067
\svnkwnk	\svnmmainfilename	..	\svntabauthor	.... 1402
\svnkwnkAuthor	..... 216, 681		\svntabdate	. 1307, 1403
\svnkwnkDate	\svnmmainurl	216, 680	\svntabfile	..... 1389
\svnkwnkdef	\svnmminute	.....	\svntabglobal	.... 1375
. 372, 373,	212, 410, 677, 1463		\svntabgroup	..... 1376
374, 375, 376,	\svnmmonth	.....	\svntable	.....
442, 443, 444,	209, 415, 676, 1460		.. 1356, 1521, 1524	
445, 488, 489,	\svnmulti@atbegininputfile		\svntablefoot	1364, 1545
490, 491, 567,	..... 789,		\svntablehead	1361, 1525
592, 593, 594,	862, 930, 1213, 1554		\svntabrev	..... 1401
595, 596, 604, 605	\svnmulti@atendinputfile		\svntabsubgroup	.. 1377
\svnkwnkdef@	... 792, 866, 1095		\svntime	..... 410
. 569, 572, 577,	\svnmulti@begininputfilehook		\svntimezone	..... 396
578, 580, 581, 582	.... 760, 775, 790		\svntimezonehour	..
\svnkwnkdef@Author	\svnmulti@endinputfilehook		214, 396, 677, 1465	
..... 580, 585	. 777, 782, 794, 796		\svntimezoneminute	.
\svnkwnkdef@Date	\svnmulti@fink@prepare		215, 396, 677, 1466	
\svnkwnkdef@LastChangedAt	..... 763, 765		\svntoday	..... 413
..... 586	\svnmulti@fink@restore		\svnurl	..... 217, 1467
\svnkwnkdef@LastChangedBy	..... 780, 783		\svnyear	.....
..... 585	\svnmulti@Gin@setfile		208, 415, 676, 1459	
\svnkwnkdef@LastChangedRevision	..... 1241, 1246			
..... 584	\svnmulti@pgf@declareimage	T		
\svnkwnkdef@Rev	..... 1253, 1255		\thesvncg	..... 911
.... 575, 583, 584	\svnnolinkurl	1413, 1416	\thesvngroup	..... 892
\svnkwnkdef@Revision				