The svn-multi Package

Martin Scharrer
martin@scharrer-online.de

CTAN: http://www.ctan.org/pkg/svn-multi

VC: https://bitbucket.org/martin_scharrer/svn-multi

Version v2.4d – 2011/08/30

1   Introduction

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

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

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

- Macros to typeset the VC information of the current source file.

- Access of all parts of the VC date.

- Formatting of author names or revision numbers.

- Definition of groups and subgroups.

- Including of the VC information of external files.

- Table of Revisions.

1.1   Scope of Keywords

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

---

[1]Subversion homepage: http://subversion.tigris.org/

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

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

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

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

2 Usage

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

2.1 Package Options

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

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

The available options are:

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

all Activates all features of the package except of the experimental ones.

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

external Controls the support for keywords from external files described in section 2.10. This needs either the external script svn-multi.pl or the autokw option.

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

groups Controls the keyword groups feature described in section 2.3.

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

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

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

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

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

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

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

autoload   Controls automatic loading of corresponding .svx files at the begin of files included using \input or \import. This avoids the need of putting an \svnexternal macro in every file just to load the .svx files created automatically by the graphics option. The option external is activated by autoload.

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

filehooks   This option loads the filehook package and installs at-begin-input-file and at-end-input-file hooks which are needed for many of the options above. While this option is enabled automatically if needed it can be also enabled manually to ensure that the file-local macros are reset to empty values at the begin of each input file. This prevents the keyword from leaking over from one file to the next. After every subfile the file-local keyword macros are also restored to the value of the parent file. A \clearpage should be added at the very end of \included files to ensure the last page is flushed out by TeX before the keyword macros are restored. Otherwise the last page might display the mainfile keyword values.

autokw   This experimental feature allows the automatically extraction of the keyword values from the hidden Subversion working copy database. The database (a text file called 'entries') is located in the hidden Subversion directory '.svn' (or '_svn' on some Windows installations) inside every directory which is under VC.

This feature makes an external script like svn-multi.pl or even keyword macros redundant as long the files are inside a Subversion working directory. However, this feature does not work if the files are exported (e.g. with svn export) or manually copied and also depends on the used version of Subversion. Only versions starting with 1.4 (with working copy format version 7) are supported. Earlier version used a different format for the entries file. Newer versions should be compatible as long the basic format is not changed again.

The experimental status will be lifted after the feature was tested using different Subversion versions on different platforms. Please do not hesitate to send error reports to the package author. Minimal examples and information about the used Subversion version and platform are very much appreciated.

This option allows the following values:

false Feature is disabled (default).

No value or **true** or **all** The keywords of all files are automatically extracted. No \svnid or \svnidlong macros or external scripts are necessary as long the files are inside a Subversion working directory and not exported.

ext Only the keywords of external files are extracted. This avoids the need of the svn-multi.pl script. The option external must be still enabled manually.

2.2 Including Subversion Keywords

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

> **\svnid**{Id}

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

This macro must be written like seen above while the order of arguments is not 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 the last dollar. This macro alone was the very first version of svnkw and is still included for fast and simple keyword typesetting.

---

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

> **\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 by 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

7

would typeset the latest revision of the same file or any subfile or declared external file included by it.

2.4   Typesetting the Keyword Values

The following macros can be used to typeset the keyword values anywhere in the document. Please note that not all LaTeX fonts have all special characters, e.g. '_' is not provided in the standard roman font. To proper typeset file names and URLs containing these letters you can use either teletype font (\texttt) or use {\urlstyle{rm}\svnnolinkurl{...}} which requires the hyperref package.

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

> **\svnrev**
> **\svndate**
> **\svnauthor**

These macros hold the keyword values of the whole document, i.e. of the most recent revision. They can be used everywhere in every file of the LaTeX document, after \usepackage{svn} of course. Please see section 2.5 how to typeset parts of the date.

> **\svnfilerev**
> **\svnfiledate**
> **\svnfileauthor**

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

> **\svncgrev**
> **\svncgauthor**
> **\svncgdate**

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

> **\svng**{⟨*group name*⟩}{⟨*key*⟩}

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

```
\svnmainurl
\svnmainfilename
```

The macro \svnmainurl and \svnmainfilename hold the URL and the filename of the main LaTeXfile 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 \svnkw{HeadURL} or \svnkw{Filename} after \svnid or \svnidlong, respectively).

```
\svnsetmainfile
```

This will declare the current file as the main LaTeX file by defining the above macros. It will automatically be called at the end of the preamble so the user normally doesn't have to use it by him- or herself as long it isn't needed in the preamble.

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

```
\svnkw{⟨keyword name⟩}
```

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

```
\textsl{Revision: \svnkw{Revision}}
URL: \url{\svnkw{HeadURL}}
```

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

If the given keyword doesn't exists 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.

New in version 2.2. Will change in future versions:

```
\ifsvnfilemodified{⟨case: modified⟩}{⟨case: not modified⟩}
\ifsvnmodified{⟨case: modified⟩}{⟨case: not modified⟩}
```

This two macro can be used to check if either the current or any file was modified after it was last checked into the repository. At the moment (v2.2) a file is marked 'modified' if there is either a '*' or 'M' after the revision number, e.g. '$Rev: 123M $'. Such an marker is automatically added for exported files ('svn export') if there where locally modified, but can also be added manually.

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

2.5 Accessing Date Values

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

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

\svnRegisterAuthor{⟨*author*⟩}{⟨*full name*⟩}

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

\svnFullAuthor{⟨*author name or macro*⟩}
\svnFullAuthor*{⟨*author name or macro*⟩}

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

---

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

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

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

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

2.9   Table of Revisions

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

\tableofrevisions

If the table is enabled a table of revision is written into a file called ⟨*main 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

Figure 1: Table formatting macros and their position.

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}{llll} \end{tabular}` group by default. They can be redefined using
        `\renewcommand{svntable}{...}{...}`.
In order to support special table packages like `tabularx` or `longtable` the *content* of this macro is written verbatim into the `.svt` file. All other table macros are written as string and will be expanded when the `.svt` is read back.

> `\svntablehead`

Holds the table head row "Name & Rev & Author & Date \\\hline".

> `\svntablefoot`

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

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

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

> `\svntabglobal`

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

> `\svntabgroup`{⟨*group name*⟩}

Formats the group name.

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

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

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

Same as for the subgroup above but for file names.

> `\svntabrev`{⟨*revision number*⟩}

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

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

> **\svntabauthor**{⟨*author username*⟩}

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

> **\svntabdate**{{⟨*year*⟩}{⟨*month*⟩}{⟨*day*⟩}{⟨*hour*⟩}{⟨*minute*⟩}{⟨*second*⟩}{⟨*TZ hour*⟩}{⟨*TZ minute*⟩}}

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

2.10 Including Keywords of External Files

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

> **\svnexternal**[⟨*group name*⟩]{⟨*file 1*⟩}{⟨*file 2*⟩}...{⟨*file n*⟩}

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

> **\svnexternalpath**{{⟨*path 1*⟩}{⟨*path 2*⟩}...{⟨*path n*⟩}}

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

Script svn-multi.pl

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

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

1. Compile LaTeX document, .aux file is generated.

2. Run `svn-multi.pl` script, `.svx` files are generated.

3. Compile LaTeX document, `.svx` files are read in.

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

```
svn-multi.pl ⟨jobname⟩
```

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

```
svn-multi.pl ⟨jobname⟩ [--group ⟨group name⟩] ⟨file(s)⟩ ...
```

The second way to use `svn-multi.pl` is to call it with a list of external files. A keyword group can be specified using the `--group` ⟨group name⟩ option which can 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, possible empty, `\svnexternal` macro is included. This allows for easy including of many external files without specifying them all inside the source file. For example `svn-multi.pl */*.jpg` (under Linux/Unix) will include the keywords of all JPG files in all subdirectories.

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

```
svn-multi.pl ⟨jobname⟩ [--group ⟨group name⟩] --fls
```

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

3 Compile Guide

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

Basic Features – Global and Group Keywords

1. Compile document with LaTeX. The `.aux` file is generated. Document global and group keyword macros are not valid yet.

2. Compile document again with LaTeX. The `.aux` file is read by `svn-multi`. Document global and group keyword macros are now valid.

Table of Revisions

1. Compile document with LaTeX. Document global and group keyword macros are calculated and written to the `.svt` file.

2. Compile document again with LaTeX. The `.svt` file is read and typeset by the `\tableofrevisions` macro.

External Files

1. Compile document with LaTeX. The `.aux` file is generated with references to the external files. Document global and group keyword macros are not valid yet. External files are not taken into account.

2. Run `svn-multi.pl` with the main base name (main file name without extension) as argument. This generates `.svx` files for each `.tex` file which used `\svnexternal`. This files contain the keywords of the external files.

3. Compile document again with LaTeX. The `.svx` files are read by `svn-multi` and the `.aux` file is updated to take the new keywords into account. Document global and group keywords macros only hold internal values.

4. Compile document again with LaTeX. The `.aux` file is read by `svn-multi`. Document global and group keyword macros are now fully valid.

4   Known Issues

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

4.1   Packet `listings` uses `\input`

Update: Newer versions of `svn-multi` avoid this issue by changing the catcodes back to normal while reading the `.svx` file. If a file ⟨*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 this packages, their internal macro which got used and acknowledge the authors/maintainers of them. Please send error reports to the author of `svn-multi` and not to the people listed below.

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

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

hyperref

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

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

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

Licence: LPPL, any version

Authors/Maintainers: Sebastian Rahtz, Heiko Oberdiek

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

graphics

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

Patched internal macros: `\Gin@setfile`

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

Licence: LPPL, any version

Author/Maintainer: David Carlisle, LaTeX3 Project

Location: CTAN: http://tug.ctan.org/pkg/graphics

pgf

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

Patched internal macros: `\pgf@declareimage`, `\pgf@image@`⟨*image name*⟩!

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

Version used: 2008/01/15 v2.00

Licence: LPPL v1.3c and GPL v2

Author&Maintainer: Till Tantau

Location: CTAN: http://tug.ctan.org/pkg/pgf

`latex`

Parts of the macro definitions of the `\tableofcontents` macros from the `article` and `book` class of standard LaTeX 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:**   LaTeX3 Project
>
> **Location:**   CTAN: http://tug.ctan.org/pkg/latex

`currfile`

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

> **Version used:**   2011/01/03 v0.3
>
> **Location:**   CTAN: http://tug.ctan.org/pkg/currfile

`filehook`

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

> **Version used:**   2011/01/03 v0.4
>
> **Location:**   CTAN: http://tug.ctan.org/pkg/filehook

6   Further Reading

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

[1] Martin Scharrer, "Version Control of LaTeX Documents with svn-multi", The PracTeX Journal, (3), 2007. URL: http://www.tug.org/pracjourn/2007-3/scharrer/

[2] Mark Eli Kalderon, "LaTeX and Subversion", The PracTeX Journal, (3), 2007. URL: http://www.tug.org/pracjourn/2007-3/kalderon-svnmulti/

[3] Uwe Ziegenhagen , "LaTeX Document Management with Subversion", The PracTeX Journal, (3), 2007. URL: http://www.tug.org/pracjourn/2007-3/ziegenhagen/