TEX Versioning System

Software for storing complete source codes of TEX documents

by David Antoš

1 Introduction

This is documentation of TEX Versioning System, software for storing complete source codes of TEX documents.

Version 1.0

(Documentation CVS ver. $Id: tvs.texi,v 1.23 2000/08/16 17:00:24 antos Exp $)

1.1 Why do you need `TVS`?

It must have happened to you. You use TEX at work. You run another distribution at home. You wrote a beautiful document at home, you came to work and got error message `I can't find file 'foo.tex'`. There is a package missing in your work installation.

Or: You wrote a beautifully typeset book several years ago. You used em-TEX on M$-DOS machine. The book should be typeset again. You have only my_book.tex and no macros you used. What to do?

To be serious, there is nothing to be done now. But if you stored the *complete* source of the documents, it didn't have to happen to you. And how to store the complete source? It's simple—use `TVS`.

Please! Please! Packing complete TEX document sources is a delicate and sensitive operation. Read the entire manual before you use `TVS` to do something important. This kind of software can be useful for you if and only if you use it correctly. I'm sorry for this but it's the fact.

1.2 What `TVS` can do?

`TVS` can store complete source codes of TEX documents. It parses logs of TEX and finds files which were used to typeset the document. `TVS` can work in three levels:

- Only document log is examined. It stores all macros, styles, included documents etc. you need.
- Document log and format log is examined. It also stores all source files needed to build up format file.
- (experimental) Moreover dvips log is parsed and used fonts are stored. This is not so useful as only `.pk` files may be obtained this way.

Files collected by `TVS` can be used on any TEX installation to build up exactly the same document. Next, let us call the product of `TVS` a *complete source*.

There are several important features of `TVS`:

- `TVS` is a Perl script, it runs wherever Perl is available (it means nearly everywhere).
- There is no difference in using of `TVS` when running LaTEX, plain, or whatever you want.
- On the destination (the place you typeset the complete source), there is no need to run `TVS`. You only need to say TEX to find files in working directory first.
- `TVS` is *not* a version control system. It cannot handle versions of your documents. There are other systems, e.g., `RCS` or `CVS`, designed to do this.

1.3 What do you need to run TVS?

Hardware and OS requirements are simple. `TVS` is a Perl script so it runs wherever Perl 5.x is available. It uses packages Getopt::Long, File::Basename, File::Copy, and File::Path. These packages should be nevertheless in standard Perl distribution.

For information on Perl for your OS check `http://www.perl.com`. There should be no problem on most Unices, in Linux, Perl is usually a part of standard distribution.

Maybe you find useful having also `make` and `Texinfo` utilities installed on your system, but `Texinfo` is only needed to produce documentation in DVI, PostScript, Info, or HTML formats which are pregenerated in `TVS` distribution. And `make` is only to generate them comfortably.

There are some requirements for your TEX source codes to be able to handle them successfully. There is a problem with including files into your source. The usual and best way to do it is to have the main source at the top of the source tree and use relative paths in all variants of `\input` command. Avoid pathological inputs like `\includegraphics ../../pictures/pict1.ps`. This should not bother you as nearly everybody does it.

Usually `TVS` is able to pack files referenced by "up-going" relative path (if it makes a conflict, `TVS` recognizes it), but you run into difficulties when re-typesetting the complete source and you have to correct all the things by hand.

2 Installation

The program `TVS` and its documentation is written and maintained by David Antoš, `xantos (at) fi.muni.cz`. (Of course substitute (at) with `@`.) Contact him when in trouble, but please read the documentation first.

The following instructions assume that Perl version >=5 is installed on your system. For information on Perl consult `http://www.perl.org`.

The homepage of `TVS` is `http://www.fi.muni.cz/~xantos/tvs`. Here you can get `TVS` either file-by-file or as `tvs.tar.gz` file. `TVS` can also be obtained on CTAN in `support/TVS`.

Because of platform independence we decided not to prepare any "instant" packages.

If you downloaded `tvs.tar.gz`, gunzip and un-tar it somewhere. If you downloaded separate files, reconstruct the original file structure anywhere you want.

On UNIX, make a link to `tvs.pl` from a directory which is in your `PATH`, e.g. `/usr/local/bin`, if your system decodes first lines of scripts to run them. In this case, it may be needed to correct the very first line of `tvs.pl` and put the right path to your Perl interpreter there. Otherwise you may still run `TVS` running `perl tvs.pl`. Install the documentation in your favorite format, where your docs-reader can find it. At least, `/usr/doc/tvs` wouldn't be a bad choice to store Post Script or HTML documentation.

Generally speaking, there is no need to install `TVS` to global directories if you are the only-one who runs it or you are not root.

Try `tvs --help`. If a TVS header and help message written, everything is OK.

If your standard Perl distribution does not contain packages Getopt::Long, File::Basename, File::Copy, and File::Path, get them from `http://www.perl.org`.

On M$ DOS/Window$ platforms the `.pl` suffix should be enough for Perl script to run, of course if Perl is correctly installed. Please do not expect me to make better support for M$ platforms as I do not run them.

On other platforms, if there are difficulties in installing or running `TVS`, please drop me a line.

3 Using TVS step by step

This chapter shows the usage of TVS and various examples. It also points out the limitations of this system, in other words what it *can* do for you (and it's quite a lot) and what it cannot. Yes, there are some unpleasant things you must do manually.

3.1 Useful commands

To make sure that TVS is correctly installed on your system, run `tvs -h`. You should obtain a short help on using TVS.

There are two useful switches you should know before starting experiments with TVS. You may increase verbosity level using `-v` or even `-vv` switch. See Section 4.2 [Global switches], page 8, for detailed information on verbosity levels.

In following examples you may also specify `--dummy` command. It means that no files are really copied and no real output is produced. But everything else looks like real operations were performed.

3.2 Basic running TVS

Now it's time to look to the `examples` directory in your distribution. It contains source file `sample.tex` we will use to demonstrate the usage of TVS.

The first step you must not forget is to TeX your source correctly. Document `sample.tex` uses `\tableofcontents`, so you must run

```
latex sample.tex
```

twice. Make sure your document is TeXed correctly. It is *very very* important.

Let's try what TVS can do now. First try to run

```
tvs sample.log
```

The most important for TVS is log file `sample.log`. You do not have to specify name with `.log` extension as it is tried to add automatically. What TVS now does? It goes through the file `sample.log` and finds all the files which are needed to build up the final document.

TVS created an output directory named `TVS-OUT`. Check its contents. In fact, on M$-DOS or M$-Windows machines this didn't work.[1] On UN*X machines, there is stupidly deep directory structure containing only one directory in another.

There is a way to avoid this. Please copy file `pathcuts.tvs` into the `examples` directory. Edit `pathcuts.tvs`, replace path `/usr/TeX/texmf` with the path to your TeX main directory.

Delete the `TVS-OUT` directory and run

```
tvs -v -p pathcuts.tvs sample
```

See what TVS does now. You may also check what is in the `TVS-OUT` directory. By `-p file` we say that `file` contains paths we want to shorten. On M$ file system, there is in fact needed to use this to cut things like `c:\` to be able to construct file names used in the output directory.

[1] Do not install Linux immediately, we will correct it in the next paragraph. ;-)

3.3 Ignoring files

As you can see, there is `sample.toc` in the output directory. We do not really need it as it can be always generated. Now let us show the way how to ignore files. Please copy file `ignpats.tvs` to the working directory. This file contains patterns which are compared to ends of files. Matching files are not copied to the output directory.

If you want to try it, delete the output directory and run

```
tvs -v -p pathcuts.tvs -i ignpats.tvs sample
```

Now files ending with `toc` are missing.

Warning: use this feature with care and only if you know what you are doing. Always make sure that files you want to ignore may be rebuilt up from the rest of the source.

3.4 Format file sources

In fact, storing the complete document source may not be enough to be able to produce the same output. We must also store format source files. Storing only the format file itself doesn't help as this file is not binary compatible in general.

The main thing you need is to know where log files of generated formats are stored. To obtain this information consult your local documentation, or you may try to find files `latex.log`, `tex.log`, `pdftex.log`, or ask your TEX administrator. It is also possible that your administrator deleted log files after generating formats. In this case ask him to regenerate format files and leave the logs untouched. This is not a good manner to delete log files of this importance. Your administrator may have even delete format file sources. In this case, ask him to reinstall them and better find a new TEX administrator.

You should make sure that formats were really generated on your local installation. They may have been generated on a binary compatible machine with other organization of TEX directories. If so, you must re-generate them. If you run a distribution which allows you to set local TEX installation root directory, this is quite possible. For example, on my texlive-4 I had to correct this.

In emergency, if you do not have root privilegies and you cannot globally make format logs correct, you still can generate formats in your own directory and use them overriding your system defaults. In this case, give your root my regards.

To store format sources, you need to tell `TVS` where logs are. It is done by `-f path` switch. Let's try to call

```
tvs -v -p pathcuts.tvs -f where_your_logs_are sample
```

You should obtain output directory similar to previous one but containing subdirectory named `FORMAT`, where complete source for format generating is stored.

If `TVS` is not able to determine format name correctly, you may force format name by adding `--fmtname name` into command line similar to previous example. You may try

```
tvs -v -p pathcuts.tvs -f where_your_logs_are --fmtname latex sample
```

3.5 Configuration file

The command lines we use are getting too long. If file `tvsrc` is located in the working directory, it is used as config file. You may also specify the config file explicitly by `-c filename` or `--config filename` switch.

Please copy file `tvsrc` to the `examples` directory. Config file has simple structure

```
<variable> <value>
```

Values set here may be always overridden by the command line. Setting values for ignore, pathcuts and format doesn't mean that files are ignored, paths cut and format sources stored by default. You must use switches `-i, -p` and `-f`, but there is no need to follow them by values. It may seem a bit strange but it is done this way in order to prevent you from doing something you do not want to. It doesn't save your keystrokes but—we hope—your nerves.

If you edit the `format` entry in `tvsrc` to the path to your format logs, you may for example use

```
tvs -i -p -f -v sample
```

to store the source with format with ignoring files and cutting paths instead of the long form without config file which goes

```
tvs -i ignpats.tvs -p pathcuts.tvs -f /usr/TeX/texmf/web2c -v sample
```

But if you use only

```
tvs -p sample
```

then no format file is examined and no files are ignored, no matter that the config file exists.

In the config file you may also specify the default output directory name with `output` entry and the default verbosity level using `verbose` with value 0, 1, or 2. These values are used always and may be overridden with `-o` and `-v`.

3.6 Limitations of TVS

Work of TVS is limited by the files which are noticed in log files. If you use anything which is done by external program, you must take special care on files you want to store. If you include graphics in Post Script, it may be included only by `\special` command and may not be noticed in log.

You must always ask yourself the main important question: *Which files—not used in TEX last run—I want to store?*

For example if you use a bibliography, your source is a `.bib` file. But in the last TEX run only sorted part of it, `.bbl` file is included. If the `.bbl` file is enough for you, it's good. But if you want to store sources for bibliography completely, you need also `.bib` file and `.bst` file, which contains bibliography style. And there is no way for TVS to know it as BibTEX is an external utility. That's why you must copy them to `TVS-OUT` by hand.

If you don't do so, you are still able to obtain exactly the same document, but you run into difficulties trying to do any modifications.

It is always better store more than you need. If you are a TEX beginner and you are absolutely not sure, it may be reasonable for you to preserve all the files from your working directory. Includes from TEX directory (e.g., `c:\tex\texmf`) are handled correctly.

3.7 Fonts

This is experimental!

Usually there is no reason to store also font source files. The exception is if your national modification of fonts changed its encoding or you use unusual fonts.

If you want to, you may run `dvips` with `-d` switch to produce a log file with list of opened files. Consult your `dvips` documentation how to do it. This switch is followed by number indicating what should be logged.

If you use `tvs -d dvips.log`, a subdirectory FONTS is created in the output directory and all files used by dvips are stored there. Only font bitmaps may be obtained this way, so this feature is not very useful. If you want to store Metafont sources, you must do it by hand.

There is no guarantee of correct work of `-d` switch as `dvips` log changed.

3.8 What to do with collected output

Currently `TVS` implements no way of packing nor extracting collected complete source. Use any way of packing this directory you like depending of the platform you run `TVS` and the platform you will want to re-typeset your work in the future.

There is no way to run TVS on the destination. You must only tell TeX

- to generate the format file if format sources were stored,
- to generate fonts if fonts were stored,
- to find files in working directory and its subdirectories at first.

To do this, consult the local documentation.

4 Reference guide to TVS commands

This chapter describes overall structure of TVS commands and describes the function of the commands in detail.

4.1 Invoking TVS

General structure of calling TVS is

```
tvs [switches] filename
```

where `filename` is a name of the main TeX log file. TVS must be run in the directory where main TeX log file is located.

Always make sure that your document is TeX-ed correctly (index is sorted, bibliography is present, pictures are included etc.)! Usually you have to run TeX several times to typeset the document (but we hope this doesn't surprise you).

4.2 Global switches

These switches may be used to change globally behavior of TVS.

`-h, --help`
: Prints short usage and version information and exits successfully. This switch doesn't need a filename as a parameter.

`-v, --verbose`
: Sets verbosity level. May used several times. There are three verbosity levels:
  - Level 0—only examined files and errors are written. This is the default.
  - Level 1—found files, copying info and errors are written.
  - Level 2—extra talkative, writes all it does.

  For level 1, use `-v` or `--verbose` once, for level 2, use it twice. You may also specify `-vv`, which is the most comfortable way to do it.

`-c file, --config file`
: Specifies the configuration file name. See Section 5.1 [Config file], page 10, for details about configuration file.

  If `--config` switch is not specified, file `tvsrc` in working directory is used as a config file if it exists. There is no need of having config file as all options may be set from command line.

`--dummy`
: Sets TVS to dummy mode. No files are changed, no files are really copied to output directory, it is only shown what has to be done. Useful if used with `--verbose`. (Inspired by similar switch in cdrecord.)

4.3 Ignoring files and cutting filenames

Using following settings we may set files which shouldn't be copied to output and set usual paths we don't want to store completely.

`-i [file], --ignore [file]`
: Sets file with patterns of filenames which we don't want to be copied to output. See Section 5.2.2 [Ignore pattern file], page 11, for details about ignore pattern file.

  If `file` name is omitted, value from config file is used. If `ignore` parameter is not set in config file, TVS terminates with an error.

`-p [file], --pathcuts [file]`
: Sets file with path patterns to cut. See Section 5.2.1 [Pathcuts pattern file], page 11, for information about file path names handling.

  If `file` name is not specified, value from config file is used. If `pathcuts` parameter in config file is not specified, `--pathcuts` parameter is silently ignored. It makes no harm to function of `TVS` and if it causes filename conflict, `TVS` stops with an error.

4.4 Output directory setting

`-o dirname, --output dirname`
: Sets the name of output directory to `dirname`. This may be only useful to override setting in config file. If config file doesn't set output directory name, default name `TVS-OUT` is used and this is created in working directory.

4.5 Storing format source files and fonts

`-f [dirname], --format [dirname]`
: Specifies directory where format log files are and packs format file sources. If `dirname` is omitted, value from config file is taken. If `format` parameter is not specified in config file, `TVS` dies with an error. See Section 3.4 [Format file sources], page 5, for details about format source files extracting. Format file name is taken from the main document log from the first line which contains `format=name`.

`--fmtname [name]`
: Specifies format name, e.g. `plain`, `latex`, etc. Useful, if format name cannot be found in log of document in form `format=name`. Specifying `--fmtname [name]` doesn't imply packing format sources. You must specify `--format`, too. This option cannot be specified in `TVS`'s config file.

`-d file, --dvips file`
: Sets name of `dvips` log file. This cannot be specified in config file. This is only experimental. See Section 3.7 [Fonts], page 6, for information on dvips log file examining and font storing. You would better not to use this feature unless you know what you are doing.

5 Config file and pattern files

In this chapter we discuss config file setting and structure and usage of files with patterns. These are used to shorten (cut) filenames to use reasonable short paths in output directory and to set files which we do not want to store at all.

5.1 Config file

We may set values for `verbose`, `format`, `ignore`, `pathcuts`, and `output` variables in config file.

If `TVS` is run with `--config` switch, it sets used config file name. If this is not used, default config file named `tvsrc` in working directory is tried. If it doesn't exist, no config file is used at all. There is no default config file location (e.g. `/etc/tvsrc` or something similar) because of the platform independence of `TVS`.

The structure of config file is quite simple. Everything after `#` and `%` characters is ignored. Lines with structure

```
<variable> <value>
```

set values of variables. There must be just one white character between the `<variable>` and the `<value>`. There may be any count of white spaces before and after this structure on the line. Lines with other structure or specifying variables not listed below are silently ignored.

For example of config file, see file `tvsrc` enclosed with the distribution.

Variables have meanings as follows:

`verbose`
: Sets the default verbosity level. Reasonable values are

  `0`
  : This is the default.

  `1, 2`
  : Set verbosity level to 1 or 2. See Section 4.2 [Global switches], page 8, for description of verbosity levels.

`format`
: Sets the name of the directory where format log files are stored. This is used only of `--format` or `-f` switch is set and may be overridden by this. You should specify `format` parameter in config file if you want to store format sources. And you should store format sources. Simply—use it.

`ignore`
: Sets the name of file with patterns used to ignore files. See Section 5.2.2 [Ignore pattern file], page 11, for information about ignoring files. This is used only if `--ignore` or `-i` switch is set and may be overridden by this.

`pathcuts`
: Sets the name of file with patterns used to shorten the stored filenames. Used only if `--pathcuts` or `-p` switch is set and may be overridden by these.

`output`
: Sets the default output directory name. May be overridden by `--output` or `-o` switch. If not set, default value `TVS-OUT` is used.

5.2 Pattern file structure

The structure of files which contain patterns is quite simple. Everything after `#` or `%` characters is ignored. A line of the file may start and end with white characters. These white characters are ignored. The remaining part of the line (if it is nonempty) is considered to be a pattern.

A pattern is treated as a Perl regular expression. It doesn't matter if you don't know what it is, all you need to know is seen from the examples. Nevertheless you may try `man perlre` or equivalent for reading documentation on you system.

Next, we'll discuss the usage of pattern files.

5.2.1 Pathcuts pattern file

This file is used if `--pathcuts` or `-p` switch is specified. An example is enclosed with the distribution and named `pathcuts.tvs`.

The file specifies list of patterns which are compared to the *beginnings* of filenames. On *first match* the matching part is cut and the rest of pattern list is not used. So it has no sense to have a pattern list

```
/usr/TeX
/usr/TeX/texmf
```

as the second entry will be never used. But sequence

```
/usr/TeX/texmf
/usr/TeX
```

may be reasonable. (It's an exercise why and when `:-)`.)

An example: Let us have pattern list containing `/usr/TeX/texmf`. If filename is

```
/usr/TeX/texmf/tex/latex/algorith/algorithm.sty
```

it matches and is cut to

```
tex/latex/algorith/algorithm.sty
```

File `inputs/my_macro.tex` doesn't match the pattern and is left untouched.

5.2.2 Ignore pattern file

This file is used when `--ignore` or `-i` switch is specified. An example named `ignpats.tvs` is enclosed.

This file specifies list of patterns which are compared to *ends* of filenames. If *any* of them matches, file is ignored and is not copied to output. Usually, suffices of files (e.g. `toc, bbl`) are used here. If you want make sure that there is a dot before `toc` or `bbl`, use `\.toc` and `\.bbl` entries in pattern file as entries are PERL regular expressions. For detailed information see `perlre(1)`.

Use this feature with care, it is usually better to have a superset of needed complete source then have *incomplete* source. There may be files which loaded (and you don't know about them) with suffices you don't expect. Use it only if you know what you are doing. You've been warned.

Appendix A Credits

I would like to thank to:

- Petr Sojka, who made me to write `TVS` and helped me very much with all the troubles and decisions we have had to make,
- Richard Botík, who lent me the “Perl Camel Book” for a fortnight,
- many others,
- Bubo, my home computer, as it surprisingly didn’t get overheated too much during the terribly hot spring of year 2000 in middle Europe.

Appendix B Dealing with bugs in TVS

Please send bug-reports to `xantos (at) fi.muni.cz` with subject `TVS bug`. Add information on platform, TeX installation and include log files which were parsed. Also please note version of TVS, both version number and CVS version code (it's the line containing `$Id: tvs.pl,v...`). You can get this information by running `tvs --help`.

If you tried to compare `.dvi` files obtained running TeX on various platforms using `TVS` complete source, please keep in mind they may differ if something depending on "the state of the computer" (especially date) was included.

Table of Contents