The mVersion package*

Michael Schmeing
michael.schmeing@googlemail.com

October 19, 2011

1 Introduction

The mVersion package is an easy way to keep track of different versions of your document. It provides a counter that can be incremented each time you compile the document. By displaying the counter on each page, (e.g. with the hyperref-package) you can see which of two versions is the newer one.

The version number is considered to consist of two parts which are separated by a dot. The first part is a fixed string that can be set by the user with `setVersion`. The second part is the build number which can be incremented by calling `increaseBuild`. The command `version` prints the complete version number, e.g. 0.1.334 with 0.1 being the fixed version number and 334 the build number.

Fixed version number and build number are automatically stored in the file *version.dat* which is created by mVersion. By calling `increaseBuild` in the preamble of your document, you can increase the build number of your document each time you compile.

2 Usage

The mVersion package provides three commands.

`version` This command prints out the version number consisting of the fixed part, followed by a dot and the build number.

`setBuild` Sets the fixed part of the version number

`increaseBuild` Each time `increaseBuild` is called, the build number that can be displayed with `version` is increased by one.

3 Implementation

We begin with a small macro `parseline` that reads a line of the form X;Y followed by one space. It then assigns X to `versionnumber` and Y to `bulidnumber`. This

*This document corresponds to mVersion v1.0, dated 2011/10/18.

macro helps reading the fixed version number and the build number from the file *version.dat*. Since LaTeX adds an additional space to each line it reads from file, we need this macro.

```
\def\parseline#1;#2 %this space is important
{
  \def\versionnumber{#1}
  \def\buildnumber{#2}
}
```

3.1 Things to do on startup

First, check if version file *version.dat* exsists.

```
\IfFileExists{version.dat}
{ }
```

If not, initialize with version number 0.0 and build number 0 and write the version file to disk. Note that the version file stores the version information in the format X;Y with X being the fixed version number and Y the build number.

```
{
\newwrite\outfile
\immediate\openout\outfile=version.dat
\immediate\write\outfile{0.0;0}
\immediate\closeout\outfile
}
```

After making sure that the version file exists, read version and build number from version file

```
\newread\versionfile
\openin\versionfile=version.dat
\read\versionfile to \versionline
\closein\versionfile
```

`versionline` no contains the version information in the X;Y–format but with an additional space following. Therefore, we let `versionline` be parsed by `pareseline`.

```
\expandafter\parseline\versionline
```

Finally, we initialize the build counter

```
\newcounter{buildcounter}
\setcounter{buildcounter}{\buildnumber}
```

3.2 Command definitions

`\version` Get the current version number.

```
\newcommand{\version}{\versionnumber.\thebuildcounter}
```

`\setVersion` `setVersion` lets you set the fixed version number. First, write the new version number and old build number to version file

```
\newcommand{\setVersion}[1]
```

```
{
\newwrite\outfile
\immediate\openout\outfile=version.dat
\immediate\write\outfile{#1;\thebuildcounter}
\immediate\closeout\outfile
```

Now re-read version file to re-read version number (not build number)

```
\newread\versionfile
\openin\versionfile=version.dat
\read\versionfile to \versionline
\expandafter\parseline\versionline
\closein\versionfile
}
```

`\increaseBuild` Increase build number by one and save new version and build number to version file. By calling `increaseBuild` in the preamble, the build number is increased each time the document is compiled.

```
\newcommand{\increaseBuild}
{
```

Increase build number

```
\stepcounter{buildcounter}
```

Save version and build number to version file

```
\newwrite\outfile
\immediate\openout\outfile=version.dat
\immediate\write\outfile{\versionnumber;\thebuildcounter}
\immediate\closeout\outfile
}
```