Context sensitve ouline elements: a manual for **coseoul**

Dynamic alteration of the document structure

September 1, 2011 VERSION 1.1

written by MICHAEL TEUBNER, August 2011

Contents

1 Introduction **1**

2 Working principle **2**

3 Use cases **2**
3.1 Collaborative work 2
3.2 Writing a thesis 3
3.3 Editing a compilation 4

4 Package options **4**
4.1 Current level 4

5 Limitations **5**
5.1 Min and max levels 5
5.2 Initial level 5
5.3 Structure warnings 5
5.4 Change levels 6

6 Developement history **6**
6.1 Inspiration 6
6.2 First attempt 6
6.3 Current Version 7
6.4 Future Developement 7

7 Acknowledgements **7**
7.1 The custom TOC 7
7.2 The name **coseoul** 8
7.3 This document 8

8 Code **9**
8.1 Complete Code 9
8.2 Commented Code 10

9 Licence **12**

1 Introduction

The commands used for document outlining in LaTeX are quite rigid. If you define a heading to be at chapter level, it will remain at that level even when you move it to a different part of your document. While this is desirable most of the time, there are some cases in which a more flexible approach may be needed, like in collaborative work, when writing comprehensive documents or when assembling a document from many different sources. This package aims at providing means of such flexible outlining.

2 Working principle

Instead of specifying exactly at what level your outline element should appear, this package provides relative commands for going up and down in the outline hierarchy. The commands provided are:

- `\levelup{<title>}`
  This command goes up one level in the document hierarchy, so depending on where you are, this will insert an outline element `<title>` at one level above the current. So if for instance written inside a section, this command will insert a new chapter.

- `\leveldown{<title>}`
  This command goes down one level in the document hierarchy, so depending on where you are, this will insert an outline element `<title>` at one level below the current. So if for instance written inside a paragraph, this command will insert a new subparagraph.

- `\levelstay{<title>}`
  This command stays at the current document hierarchy level, so depending on where you are, this will insert an outline element `<title>` at the same as the current level. So if for instance written inside a subsection, this command will insert a new subsection.

- `\levelmultiup{<title>}{<levels>}`
  This goes up multiple levels in the document hierarchy, so depending on where you are, this will insert an outline element `<title>` at a level differing by `<levels>` from the current level. So if for instance writte `\levelmultiup{title}{3}` inside a paragraph, this command will insert a new section. This may be used for situation in which level skipping is required. While downwards you will go only one level, upwards a skip of more than one may be required, for instance if you are in the last subsubsection of a passage of your text and as you want to start a new passage you need to insert a section.

3 Use cases

3.1 Collaborative work

Suppose you are writing a complex document where you collaborate with others (a book on Mexican cacti, a documentation for your extensive software project or a term paper). Hopefully you all could agree to use LaTeX, but still problems arise as you only see each other about once a month. So during this month, all of you work on their respective parts, but when meeting you decide to swap parts, raise the importance of some and lowering the importance of others. Usually this would require rewriting outline elements, but with this package you only need to move stuff to the right place and probably changing one or two outline elements, all the others change accordingly as they are context sensitive, which may save you a lot of work.

Lets say Alice and Bob are writing on that cactus book. Alice, touring northern Mexico, wrote this:

```
\documentclass{scrartcl}
\usepackage{coseoul}
\begin{document}
\levelstay{Corynopuntia}         % main
\leveldown{Grusonia aggeria}     % sub
\levlestay{Grusonia agglomerata} % sub
\levlestay{Grusonia bulbispina}  % sub
\end{document}
```

While Bob, scouting southern Mexico, wrote this:

```
\documentclass{scrreprt}
\usepackage{coseoul}
\begin{document}
\levelstay{Grusonia}           % main
\leveldown{Grusonia bradtiana} % sub
\levelup{Marenopuntia}         % main
\levelstay{Grusonia marenae}   % sub
\end{document}
```

Ideally, of cause, they would store their content including outline in seperate files, but this should only be a small example. Alice, writing an article type document, would usually use sections top level outline element, while Bob, who is writing a report style document, would use sections. When combining their work either Alice (all one level up) or Bob (all one level down) would have to rewrite their outline. Instead, Bob only changes his first element to \levleup and they are ready to go!

```
\documentclass{scrartcl}
\usepackage{coseoul}
\begin{document}
% Alices contribution
\levelstay{Corynopuntia}         % main
\leveldown{Grusonia aggeria}     % sub
\levlestay{Grusonia agglomerata} % sub
\levlestay{Grusonia bulbispina}  % sub
% Bobs contribution
\levelup{Grusonia}             % main, only changed command
\leveldown{Grusonia bradtiana} % sub
\levelup{Marenopuntia}         % main
\levelstay{Grusonia marenae}   % sub
\end{document}
```

3.2 Writing a thesis

Suppose you are writing your master thesis, say on *mobility management in Germany*. You are writing for quite some time and interviewed quite some people in different cities on the topic. As you go, you realize that your initial outline, which made perfect sense at the time of writing, now seems odd. Not that it is wrong, but some parts are condensed now and should be subordinate to others, some portions are to be moved, recombined, edited or deleted, but this will mess up your outline. If you considered that something like this might happen, you used this package and have relatively little to do:

- to subordinate, just change `\levelstay` to `\leveldown`
- move portions to a new, more appropriate position. No matter what level they are now on, worst case is you have to adjust it's first outline command and that of the next portion
- recombination is really like moving, two changes at most per portion, however long or complex

3.3 Editing a compilation

Suppose you are an editor, responsible for some proceedings. The guy responsible for writing a template that any participant should use was ill, now everyone submitted documents outlined to their individual liking: structured in sections or chapters, some mistaking this for a book and using parts. Now you are left with the mess, but fortunately, they all used **coseoul**, so you `\include` all their documents and you are (almost) done!

4 Package options

There are no *real* options at this time.

4.1 Current level

While document classes defining chapters are initialized at that level, classes not defining chapters are initialized at section level.

Normally, you would not want to set `\currentlevel` manually. However, you might want to start a document with a `\part`, to achieve that you can do the following:
`\setcounter{currentlevel}{<level>}` where <level> can be:

7	for part level
6	for chapter level
5	for section level
4	for subsection level
3	for subsubsection level
2	for paragraph level
1	for subparagraph level

But be aware: if you set the level to chapter in a document not defining those, and immediately after use `\levelstay`, you will produce an error.

There will probably a function with named arguments in the future (see 5).

5 Limitations

5.1 Min and max levels

It would be desirable to specify minimum and maximum levels to use. Most people will not want to use parts, paragraphs or even subparagraphs. Right now, all levels are uses, with no chance of influence. I'm thinking of something like:

```
\usepackage[minlevel=subsubsection, maxlevel=chapter]{coseoul}
```

5.2 Initial level

Document classes with chapters are initialized at chapter level, classes without it are initialized at section level. There should be a package option for that:

```
\usepackage[initlevel=section]{coseoul}
```

5.3 Structure warnings

When you use `\leveldown` are on the lowest level or use `\levelup` on the highest level, the new element will still be set on the same level, as there are no lower or higher levels which could be used. As this will possibly compromise your structure, there should be an optional warning for such occurrence. So something like

```
...
\usepackage[warningstop, warningsbottom]{coseoul}
...
\levelup{part}
\levelup{above part}
...
\levelup{subparagraph}
\levelup{below subparagraph}
```

should result in:

I part
II above part **(Warning: level above top)**
...

subparagraph
below subparagraph (Warning: level below bottom)

Also entries in the log file would be nice.

5.4 Change levels

Manually changing the level is possible, but not comfortable, there should really be some command like:

`\setcurrentlevel{subsection}`.

6 Developement history

6.1 Inspiration

The package is based on this question on `tex.stackexchange.com`. The user Speldosa wanted to know if such flexible behaviour was possible, and so I started the development. While the first version was quite straightforward and complicated, the big number of lines of code made me think about my approach and totally redesign it.

6.2 First attempt

The very first version, although working yet, was in a well human readable but not elegant form. It consisted of many, many `\ifthenelse` constructs which also had to be in the right order. The current level was denoted by a string and therefore had to be redefined quite often. As no standard level was defined, one of the absolute commands like `\newchapter` had to be used. `\levlemultiup` did not exist yet, there also absolute commands had to be used.

```
...
\newcommand{\currentlevel}{}

\newcommand{\levelup}[1]%
{ \ifthenelse{\equal{\currentlevel}{c}}%
  {\chapter{#1}\renewcommand{\currentlevel}{c}}{}%
  \ifthenelse{\equal{\currentlevel}{s}}%
  {\chapter{#1}\renewcommand{\currentlevel}{c}}{}%
...
\newcommand{\newchapter}[1]%
  {\chapter{#1}\renewcommand{\currentlevel}{c}}
...
```

6.3 Current Version

When I tried to expand the first version to all outline elements and also a `\levlemultiup`, I realized that this would be $(3 \cdot 2 + 7 \cdot 5) \cdot 2 = 82$ new lines of code, and that for achieving relatively little, so I decided to rewrite the code from scratch. The indicator for the current level (`\currentlevel`), formerly containing strings, is now a counter, thus operations on levels now are arithmetic operations instead of string rewrites. Instead of defining separate commands for each task (`\levleup` and the others), they are now derived from a general command. `\currentlevel` is never set to chapter if used in a `\documentclass` not defining it. `\documentclass`es defining chapters are initialized as such, while those which do not are initialized at section level.

```
...
\newcommand{\chex}{}
\ifthenelse{\isundefined{\chapter}}%
  {\renewcommand{\chex}{N}}{\renewcommand{\chex}{Y}}
...
\newcounter{currentlevel}
...
\newcommand{\findnewlevel}[1]%
{ \addtocounter{currentlevel}{#1}%
...
\newcommand{\levelchange}[2]%
{ \findnewlevel{#2}%
  \ifthenelse{\value{currentlevel} = 1}{\subparagraph{#1}}{}%
...
\newcommand{\levelup}[1]{\levelchange{#1}{1}}
...
```

6.4 Future Developement

The current version, while being fully functional, has several shortcomings (see 5). I plan on adding more features, but I don't know when or to what extend, so better not wait for changes. The current version is also maintained by KRAKEN at BitBucket.org

7 Acknowledgements

First of all, thanks to MAHOK, you first made me aware that something awesome like LaTeX existed.

7.1 The custom TOC

The table of contents was modelled on an example from ELKE and MICHAEL NIEDERMAIR: "LaTeX. Das Praxisbuch". Thanks for the example and the book as a whole, it's what got me started with LaTeX.

7.2 The name coseoul

I decided to go with the first two letters of each word in the phrase ’Context Sensitive Outline Elements’, which would be ’CoSeOuEl’. Four vocals in a row was too much for my taste, so my alternate ideas were ’consul’, ’console’, or something with ’Seoul’. As I’m not too much of a fan of the Romans nor want to mislead anyone into thinking this was some kind of command line tool, I decided to go with the Seoul idea. Although I have never been there, what better reason is there to visit Seoul than to see the city I named my package after!

7.3 This document

This document does *not* use any of the features of **coseoul**, as you should be able to build this documentation on your own. So you can install the package *after* you built and read the manual.

8 Code

8.1 Complete Code

```
\ProvidesPackage{coseoul}
\RequirePackage{ifthen}

\newcommand{\chex}{}

\newcounter{currentlevel}
% part          = 7   chapter        = 6   section      = 5
% subsection    = 4   subsubsection  = 3   paragraph    = 2
% subparagraph  = 1

\ifthenelse{\isundefined{\chapter}}%
  {\renewcommand{\chex}{N}\setcounter{currentlevel}{5}}%
  {\renewcommand{\chex}{Y}\setcounter{currentlevel}{6}}

\newcommand{\findnewlevel}[1]% uppity (-1 down, 0 stay, 1 up, 2-6 multiup
{ \addtocounter{currentlevel}{#1}%
  \ifthenelse{\equal{\chex}{N}}%
  { \ifthenelse{\value{currentlevel} = 6}%
    { \ifthenelse{#1 > 1}{\addtocounter{currentlevel}{1}}%
        {\addtocounter{currentlevel}{-1}}}{}}{}%
  \ifthenelse{\value{currentlevel} < 1}{\setcounter{currentlevel}{1}}{}%
  \ifthenelse{\value{currentlevel} > 7}{\setcounter{currentlevel}{7}}{}%
}

\newcommand{\levelchange}[2]% title, uppity
{ \findnewlevel{#2}%
  \ifthenelse{\value{currentlevel} = 1}{\subparagraph{#1}}{}%
  \ifthenelse{\value{currentlevel} = 2}{\paragraph{#1}}{}%
  \ifthenelse{\value{currentlevel} = 3}{\subsubsection{#1}}{}%
  \ifthenelse{\value{currentlevel} = 4}{\subsection{#1}}{}%
  \ifthenelse{\value{currentlevel} = 5}{\section{#1}}{}%
  \ifthenelse{\value{currentlevel} = 6}{\chapter{#1}}{}%
  \ifthenelse{\value{currentlevel} = 7}{\part{#1}}{}%
}

\newcommand{\levelup}[1]{\levelchange{#1}{1}}
\newcommand{\leveldown}[1]{\levelchange{#1}{-1}}
\newcommand{\levelstay}[1]{\levelchange{#1}{0}}
\newcommand{\levelmultiup}[2]{\levelchange{#1}{#2}} %title, uppity
```

8.2 Commented Code

This section is mainly for people like me, who have never or seldomly created packages. So if you are a *deep magic wizard*, please don't take offence in my simplistic explanations.

```
\ProvidesPackage{coseoul}
\RequirePackage{ifthen}
```

The first line tells LaTeX that this file provides the package **coseoul**. Note that the file name doesn't have to coincide with the package name, so foo.sty may provide a package bar. The second line states that the package **ifthen** is required for this to work. So such commands state dependencies on other packages.

```
\newcommand{\chex}{}

\newcounter{currentlevel}
% part          = 7   chapter       = 6   section     = 5
% subsection    = 4   subsubsection = 3   paragraph   = 2
% subparagraph  = 1

\ifthenelse{\isundefined{\chapter}}%
  {\renewcommand{\chex}{N}\setcounter{currentlevel}{5}}%
  {\renewcommand{\chex}{Y}\setcounter{currentlevel}{6}}
```

First, a command `\chex` is initialized as empty, then a counter is initialized (as zero). Then it is checked, if currently a command `\chapter` is defined. If it is not, `\chex` is set to 'N' and the document is initialized at section level. If it is, then `\chex` is set to 'Y' and the document is initialized at chapter level.

```
\newcommand{\findnewlevel}[1]% uppity (-1 down, 0 stay, 1 up, 2-6 multiup
{ \addtocounter{currentlevel}{#1}%
```

a new command `\findnewlevel` with one parameter is defined. The comment explains which values it will take for which command. The number given in the argument is then added to `\currentlevel`, thus changing the active level.

```
  \ifthenelse{\equal{\chex}{N}}%
  { \ifthenelse{\value{currentlevel} = 6}%
    { \ifthenelse{#1 > 1}{\addtocounter{currentlevel}{1}}%
        {\addtocounter{currentlevel}{-1}}}{}}{}%
```

Here we have a triple nested `\ifthenelse` statement. The first two are used to check whether we are at chapter level (currentlevel = 6) although chapter is undefined (`\chex` = 'N'). If this is the case, then dependant on whether we move level up or down (1 or -1), the `\currentlevel` counter increased or decreased by one.

```
  \ifthenelse{\value{currentlevel} < 1}{\setcounter{currentlevel}{1}}{}%
  \ifthenelse{\value{currentlevel} > 7}{\setcounter{currentlevel}{7}}{}%
```

The level may range from 1 to 7. Therefore values outside that range are set to the respective nearest value. This may cause undesired effects (see 5)

```
\newcommand{\levelchange}[2]% title, uppity
{ \findnewlevel{#2}%
```

A new command `\levelchange` is defined, which has two arguments, the title for the new outline element and the desired change in level (labelled ’uppity’). Then the previously defined `\findnewlevel` is called with the desired change in level.

```
  \ifthenelse{\value{currentlevel} = 1}{\subparagraph{#1}}{}%
  \ifthenelse{\value{currentlevel} = 2}{\paragraph{#1}}{}%
  \ifthenelse{\value{currentlevel} = 3}{\subsubsection{#1}}{}%
  \ifthenelse{\value{currentlevel} = 4}{\subsection{#1}}{}%
  \ifthenelse{\value{currentlevel} = 5}{\section{#1}}{}%
  \ifthenelse{\value{currentlevel} = 6}{\chapter{#1}}{}%
  \ifthenelse{\value{currentlevel} = 7}{\part{#1}}{}%
```

According to what value `\currentlevle` was changed by `\findnewlevel`, a new outline element is inserted.

```
\newcommand{\levelup}[1]{\levelchange{#1}{1}}
\newcommand{\leveldown}[1]{\levelchange{#1}{-1}}
\newcommand{\levelstay}[1]{\levelchange{#1}{0}}
\newcommand{\levelmultiup}[2]{\levelchange{#1}{#2}} %title, uppity
```

Four commands are defined for easily changing the outline level. These are pure convineance, as you could also use `\levelchange` directly with the appropriate parameters.

9 Licence

coseoul.tex & **coseoul.pdf**
Copyright 2011 M. Teubner

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in http://www.latex-project.org/lppl.txt and version 1.3 or later is part of all distributions of LaTeX version 2005/12/01 or later.

This work has the LPPL maintenance status ‘maintained’.

The Current Maintainer of this work is M. Teubner.

This work consists of the files **coseoul.sty**, **coseoul.tex**, **cosexamp.tex** and the derived files **coseoul.pdf** and **cosexamp.pdf**.