

The SKB User Guide

Sven van der Meer

Wednesday 4th August, 2010

Outline

- 1 The L^AT_EX SKB Package
 - The Idea
 - The Story
 - The underlying Concept

- 2 User Manual
 - Getting Started
 - Headings, Files, Figures, Slides, Paths, Acronyms and References
 - Other usefull Macros

- 3 Examples
 - A simple Article

Outline

- 1 The L^AT_EX SKB Package
 - The Idea
 - The Story
 - The underlying Concept
- 2 User Manual
 - Getting Started
 - Headings, Files, Figures, Slides, Paths, Acronyms and References
 - Other usefull Macros
- 3 Examples
 - A simple Article

The L^AT_EX SKB Package

This package provides macros that help to build a repository for long living documents. It focuses on structure and re-use of text, code, figures etc. The basic concept is to first separate structure from content (i.e. text about a topic from the structure it is presented by) and then separating the content from the actual published document, thus enabling easy re-use of text blocks in different publications (i.e. text about a protocol in a short article about this protocol as well as in a book about many protocols); all without constantly copying or changing text. As a side effect, using the document classes provided, it hides a lot of LaTeX from someone who just wants to write articles and books.

The Intent

- Maintain repository for Long-living Documents using L^AT_EX
- Provide easy access to 'stuff': text, figures, code examples, ...
- Reuse 'stuff' rather than copy'n paste it all the time
- Allow easy creation of different documents using the same 'stuff'
- Limit L^AT_EX specific code, simplify the interface between User and L^AT_EX

Applicability and Side Effects

The SKB can be helpful if you

- Need to organise large amount of 'stuff'
- Want to reorganise 'stuff'
- Plan to maintain a personal repository

The SKB has side effects (and cannot do coffee for you either)

- Separating content from structure makes content (almost) context-free
- Changes in content effect (potentially) multiple documents of different type
- Versioning and change logging becomes (almost) impossible
- Your style of writing documents might need to change
- Cross-references between parts of 'stuff' need to be used carefully

Outline

- 1 The L^AT_EX SKB Package
 - The Idea
 - **The Story**
 - The underlying Concept
- 2 User Manual
 - Getting Started
 - Headings, Files, Figures, Slides, Paths, Acronyms and References
 - Other usefull Macros
- 3 Examples
 - A simple Article

The Story, short version

After 10 years writing documents...and not maintaining them properly

- My 'stuff' got distributed in multiple dimensions
- Some hidden in emails, on shared discs, external document systems, web sites
- Finding anything became almost impossible
- Tried many tools, none really helped to organise my 'mess'

Early 2009, I decided to reorganise *everything*

- Well-defined set of tools/applications: PHP, L^AT_EX, B_IB_TE_X, Inkscape, SQLite, Java
- Then I started writing the L^AT_EXpart for the SKB
- This now became the SKBpackage

The Story, long version, Part I

The Problems I was facing

- Ideas/concepts hidden and inaccessible
- Many documents in many different formats
- Reuse of ‘stuff’ requires lots of work, and involves copy and paste
- Many different versions of very similar ‘stuff’, no time for maintenance
- Same for text, figures, examples, tables, lists, acronyms, references; all created ‘stuff’

The Story, long version, Part II

Solution: Create a (unified) document repository

- Then use it as source to generate documents
- Leave text, heading text, figures etc. in the repository
- Helps to keep information updated (no more copy and paste)
- Only a few rules, little effort for maintenance

An Example from 2009 (first version of the SKB)

- Moved 4 lecture notes, 2 presentations, 1 book chapter, 2 books and 4 articles
- Created 1,314 files in 87 folders, but:
 - Removed about 100 pages of duplicated 'stuff'
 - Removed many many duplicated figures
 - Also found tons of errors from re-using already reused 'stuff'

Outline

- 1 The L^AT_EX SKB Package
 - The Idea
 - The Story
 - The underlying Concept
- 2 User Manual
 - Getting Started
 - Headings, Files, Figures, Slides, Paths, Acronyms and References
 - Other usefull Macros
- 3 Examples
 - A simple Article

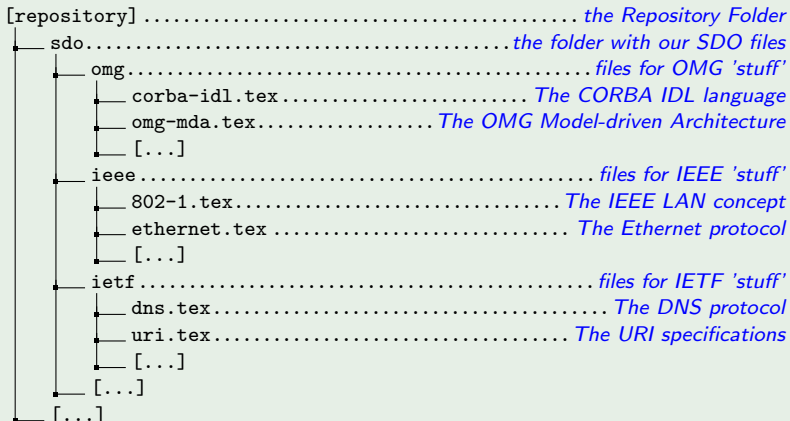
The Concept: Separate Things

Separate as much as you can (but not more)

- Separate content of a document from its structure
 - Identify small, coherent blocks of information
 - Put them in a separate folder: the repository
- Separate the different parts of a document
 - Documentation reason (publish) from repository
 - Code, figures, examples, acronyms, references

Separate Content from Structure

Example



Separate Parts of the Document, Part I

- Create a folder for data
 - Acronyms, References, ...
- Create a folder for figures

Bibliography, Acronyms and Figures

```

[root] ..... for instance /doc
├── database ..... folder for all sorts of data for the repository
│   ├── latex ..... data in LATEX, such as our acronyms
│   ├── bibtex ..... BIBTEX database
│   └── [...]
├── figures ..... folder for figures, my sources are SVG & PDF
│   └── [...]
└── repository ..... folder for the text files
    └── [...]
  
```

Separate Parts of the Document, Part II

Publications and Content

- Now look into published documents, there are different types
 - Articles, Books, Lecture Notes, Presentations, ...

The Folder `publish`

```
[your repository root] ..... path to your repository, like /dev/documents
├── [...]
├── publish ..... folder for published documents
│   ├── articles ..... ...such as articles
│   ├── books ..... ...or books
│   ├── lecture_notes ..... ...or lecture notes for computer science
│   ├── presentations ..... ...or general presentations
│   └── [...]
└── [...]
```

Separate Parts of the Document, Part III

Publications: Articles

- Now look into articles, there are a few of them, i.e.
 - Object Naming (`naming.tex`)
 - Object Models (`object-models.tex`)
 - Protocols (`protocols.tex`)
- And a separate folder with the links to repository files (`tex`)

The Folder `publish/articles`

```

articles.....our articles
├── naming.tex..... the file creating an article on naming
├── object-models.tex..... the file creating an article on object-models
├── protocols.tex..... the file creating an article on protocols
├── tex..... a folder containing the tex files that include our content
│   ├── naming.tex..... the file including all content for naming
│   ├── object-models.tex..... the file including all content for object-models
│   └── protocols.tex..... the file including all content for protocols
  
```


Outline

- 1 The L^AT_EX SKB Package
 - The Idea
 - The Story
 - The underlying Concept
- 2 User Manual
 - Getting Started
 - Headings, Files, Figures, Slides, Paths, Acronyms and References
 - Other usefull Macros
- 3 Examples
 - A simple Article

The SKB Distribution

distribution contents

[start folder]	
├── doc	<i>The generated PDFs and User Guide Sources</i>
│ ├── [user-guide]	<i>Sources for the User Guide</i>
│ ├── skb.pdf	<i>The generated Documentation</i>
│ └── skb-guide.pdf	<i>The User Guide only</i>
├── run	<i>The generated Class and Style Files</i>
│ ├── skb.cfg	<i>The global Configuration File</i>
│ ├── skb.sty	<i>The Style File</i>
│ └── *.cls	<i>The Class Files</i>
├── source	<i>The Source files</i>
│ ├── skb.dtx	<i>Documented \LaTeX Source File</i>
│ ├── skb.ins	<i>The \LaTeX Installer File</i>
│ └── *.txt	<i>Manifest, Licence, Todo List and History as plain Text</i>

Installation

First

- Go to your L^AT_EX distribution

Classes and Style Files

- Create a folder `tex/latex/skb`
- Copy `run/*` into that folder (`tex/latex/skb`)
- Update filename database of your L^AT_EX distribution

Documentation

- Create a folder `doc/latex/skb`
- Copy `doc/*` into that folder (`doc/latex/skb`)

Source

- Create a folder `source/latex/skb`
- Copy `source/*` into that folder (`source/latex/skb`)

Rebuilding the SKB

Class/Style Files and Documentation

```
#Rebuild Class and Style files
$cd run; latex ../source/skb.ins
-> creates: skb.cfg, skb.sty, skbarticle.cls, skbbook.cls,
    skbbeamer.cls, skblncsbeamer.cls and skblncsppt.cls

#Rebuild Documentation
$cd doc
$pdflatex ../source/skb.dtx           # repeat twice
$pdflatex user-guide/user-guide       # repeat twice
$pdflatex user-guide/ug-slides-anim   # repeat twice
$pdflatex user-guide/ug-slides-noanim # repeat twice
$pdflatex user-guide/ug-slides-notes  # repeat twice

# remove all files except the PDFs for cleanup
```

Configuration: Options

<code>root</code>	Sets the root path of the SKB. Everything that the SKB processes should be located below the root.	<code>/doc</code>
<code>pub</code>	Sets the path for the published documents.	<code>publish</code>
<code>rep</code>	Sets the path for the repository documents.	<code>/repository</code>
<code>fig</code>	Sets the path for figures.	<code>/figures</code>
<code>slideshow</code>	Sets the path for the slides.	<code>/transparencies</code>
<code>acr</code> , <code>acrfile</code>	The SKBuses the acronym package and these two macros detail the directory (<code>acr</code>) and the file (<code>acrfile</code>) where the acronyms can be found.	<code>acr:</code> <code>database/latex</code> <code>acrfile:</code> <code>acronym</code>
<code>bib</code> , <code>bibfile</code>	These two macros detail the directory (<code>bib</code>) and the main file (<code>bibfile</code>) where bibliographic information (BIB _T E _X database) can be found.	<code>bib:</code> <code>database/bibtex</code> <code>bibfile:</code> <code>bibliography</code>

Configuration: \skbconfig

Example (variations for \skbconfig)

```
%default content of skb.cfg
\skbconfig[
  root=/doc,fig=figures,sli=slides
  acr=database/latex,acrfile=acronym,
  bib=database/bibtex,bibfile=bibliograhpy,
  rep=repository,pub=publish
]{skb.cfg}
```

```
%using relative path for root and no directory structure
\skbconfig[
  root=.,rep=,pub=,fig=,sli=,
  acr=,acrfile=acronym,
  bib=,bibfile=bibliograhpy
]{myfile.tex}
```

Configuration: View Options Used

- Macro that can be used to show configuration options
- Shows the Change Log and actually used options
- Uses package warning to printout information
- Automatically called at the end of processing main document

Example (Root path Change Log and Option)

```
(skb)                Change log:
(skb)                - root = skb.sty, ug-slides-noanim.tex

(skb)                Last set Path/File Options:
(skb)                - file root = user-guide/
```

Creating a Directory Structure

The Directory Structure used in this Guide

```

[your repository root] ..... path to your repository, like /dev/documents
├── database ..... folder for all sorts of data for the repository
│   ├── latex ..... this is were LATEX data will be collected, such as our acronyms
│   └── bibtex ..... folder for all BIBTEX reference files
├── figures ..... folder for figures, my sources are SVG & PDF
├── publish ..... folder for published documents
│   ├── articles ..... ...such as articles
│   ├── books ..... ...or books
│   ├── lecture_notes ..... ...or lecture notes for computer science
│   └── presentations ..... ...or general presentations
└── repository ..... folder for the text content

```


Outline

- 1 The L^AT_EX SKB Package
 - The Idea
 - The Story
 - The underlying Concept
- 2 User Manual
 - Getting Started
 - Headings, Files, Figures, Slides, Paths, Acronyms and References
 - Other usefull Macros
- 3 Examples
 - A simple Article

Headings and Files

skbheading

```
\skbheading{My Heading}
```

skbinput

```
1 \skbinput{myfile}
2 \skbinput[from=rep]{myfile}
3 \skbinput[from=pub]{myfile}
4 \skbinput[level=chapter]{myfile}
5 \skbinput[from=pub,level=chapter]{myfile}
6 \skbinput[from=pub]{test/myfile}
```

- from: pub, rep, fig, sli
- level: book, part, title, chapter, section, subsection, subsection

Figures

Classic L^AT_EX Example

```
\begin{figure}\begin{center}
  \resizebox{\textwidth}{!}{
    \includegraphics[width=\textwidth]{../figures/myfig}}
  \caption{My Figure}\label{myfig}
\end{center}\end{figure}
```

skbfigure

```
1 \skbfigure{myfig}
2 \skbfigure[figure,center]{myfig}
3 \skbfigure[figure,center,width=\textwidth]{myfig}
4 \skbfigure[figure,center,
5     caption=My Figure,label=myfig]{myfig}
```

Options for skbfigure

<code>width</code>	Set the width to be used with <code>resizebox</code> and <code>includegraphics</code> .
<code>height</code>	Set the height to be used with <code>resizebox</code> and <code>includegraphics</code> .
<code>center</code>	Use <code>center</code> environment.
<code>figure</code>	Use <code>figure</code> environment.
<code>position</code>	The position to be used within <code>figure</code> environment. This option will be ignored if not combined with <code>figure</code> .
<code>caption</code>	The caption to be used. Ignored if the option <code>figure</code> is not used.
<code>label</code>	The label to be used. Ignored if the option <code>figure</code> is not used.
<code>multiinclude</code>	The label to be used. Ignored if the option <code>figure</code> is not used.

Slides

Use `\skbslide` and `\skbslidecite` to integrate slides from outside the L^AT_EX universe

- Load PDF slides with or without L^AT_EX annotations
- Load the PDF and/or the L^AT_EX from any SKB known path
- Add standardised citation for the slides to the annotation

Example with Microsoft Powerpoint

- Take a Powerpoint presentation and print all slides into a 4x3 format, individual PDFs
- Write or reuse existing annotations using L^AT_EX (and BIB_TE_X etc.)
- Produce handouts using the SKB and present the slides using Powerpoint

Slides

skbslide

```
1 \skbslide{myslides/slide1}{}
2 \skbslide{myslides/slide2}{}\clearpage
3 \skbslide[annotate]{myslides/slide3}{}
4 \skbslide[annotate,notefrom=rep]
5     {myslides/theme1}{text/theme1}
6 \skbslide[annotate,notefrom=rep,slidefrom=rep]
7     {text/theme2}{text/theme2}
```

skbslidecite

```
1 \skbslidecite{Slide}{\cite{tanenbaum-andrew:book:2003}}
2 \skbslidecite{Notes}{\cite{standard:IETF:RFC:1155}}
```

Commands to access Path and Filenames directly

- `\skbfileroot` – the root directory
- `\skbpathroot` – with 1 argument, the root-dir/#1
- `\skbfileacr` – the acronym directory and file
- `\skbfilebib` – the reference directory and file
- `\skbpathbib` – the reference directory
- `\skbfilerep` – with 1 argument, the repository-dir/#1
- `\skbfilepub` – with 1 argument, the publish-dir/#1
- `\skbfilefig` – with 1 argument, the figure-dir/#1
- `\skbfilesli` – with 1 argument, the slide-dir/#1

Example

```
\skbfileroot{examples/skbem}  
\skbfilefig{dirtree/complete}
```

Acronyms and References

Load Acronym Database

- `\skbacronyms` will load the Acronym Database
- Should be called right at the place you want to have the list of acronyms printed

Load References

- `\skbbibtex` will load the reference list
- Should be called before starting with the document

Outline

- 1 The L^AT_EX SKB Package
 - The Idea
 - The Story
 - The underlying Concept
- 2 User Manual
 - Getting Started
 - Headings, Files, Figures, Slides, Paths, Acronyms and References
 - Other usefull Macros
- 3 Examples
 - A simple Article

Emphasising Text

L^AT_EX example for skbinput

Use `\cmd{\skbem}` to produce `\skbem[bold]{bold}`,
`\skbem[italic]{italic}` or `\skbem[code]{type writer}` text.

The example above shows the macro `\skbem[code]{skbem}` with
the option `\skbem[italic]{bold}` and `\skbem[bold]{italic}`.

And the result

Use `\skbem` to produce **bold**, *italic* or type writer text.

The example above shows the macro `skbem` with the option *bold*
and **italic**.

Lists

Two new environments for lists

- `\skbnotelist` for replacing `itemize`
- `\skbnoteenum` for replacing `enumerate`

Both do nothing but changing `itemsep` and `parskip`

- Set to 0
- Same effect as `\tightlist` in the `memoir` package

Both environments do *nothing* if the `memoir` package is used!

Styles and Support for Listings

- A few pre-defined styles provided
- Most of them set a small typewriter font
- Some provide for line numbers
- `\lstdefinestyle` used to reset style to basic parameters

Example (use within normal text)

```
\lstinputlisting[style=generic,language=TeX]{example}  
\lstinputlisting[style=generic]{\skbfileroot{example-toc}}
```

Example (use for beamer frames)

```
\lstinputlisting[style=beamer-example]{example}  
\lstinputlisting[style=beamer-exampleLN]{example}
```

Macros for PDF Info

Macros to set information, call them anywere in your document:

- `\skbtitle` – the title
- `\skbauthor` – the author
- `\skbsubject` – the subject
- `\skbkeywords` – the keywords
- `date` – will be set automatically to today

SKB automatically calls the `\skbpdfinfo` macro at the end of the main document

Optional Text – Versions and Optional

- Supports the packages version and optional
- Comes with 6 defined modes, automatically configured
 - text – normal text, i.e. in an article
 - slide – slides, i.e. beamer frames
 - note – annotated slides, i.e. beamerarticle
 - anim/noanim – animated and non-animated slides/frames
 - memoir – used when memoir package is loaded

Example

```

\opt{text}{The following example}\opt{note}{This slide}%
  shows some examples for optional text (optional package)

\begin{skbmodememoir}Only with memoir\end{skbmodememoir}

\opt{anim}{\skbfigure[multiinclude=+-]{myfigure}}
\opt{noanim}{\skbfigure[width=\textwidth]{myfigure}}

```

Outline

- 1 The L^AT_EX SKB Package
 - The Idea
 - The Story
 - The underlying Concept
- 2 User Manual
 - Getting Started
 - Headings, Files, Figures, Slides, Paths, Acronyms and References
 - Other usefull Macros
- 3 Examples
 - A simple Article

An Example Article – T_EX code

Example (start of the tex file)

```
\documentclass{skbarticle}

\begin{document}
  \author{Sven van der Meer}
  \title{Protocols, Formats and Communication Services}
  \maketitle
  \tableofcontents*
  \bigskip
```


An Example Article – T_EX code

Example (loading initial text)

```
\skbinput[from=rep]{sota/protocols}
```

Example (a section called Introduction)

```
\section{Introduction}
  \skbinput[from=rep,level=subsection]
    {sota/protocols/data_encoding}
  \skbinput[from=rep,level=subsection]
    {sota/protocols/message-formates}
  \skbinput[from=rep,level=subsection]
    {sota/protocols/protocols}
  \skbinput[from=rep,level=subsection]
    {sota/protocols/protocol-services}
```

An Example Article – T_EX code

Example (more sections)

```
\skbinput[from=rep,level=section]{sdo/omg/corba-giop}  
\skbinput[from=rep,level=section]{sdo/ietf/snmp-protocol}  
\skbinput[from=rep,level=section]{sdo/itu/x700-cmip}  
\skbinput[from=rep,level=section]{sdo/w3c/http}
```

Example (finishing the tex file)

```
\end{document}  
\endinput
```

An Example Article – The produced TOC

Example

1	Introduction	1
1.1	Data Encoding	2
1.2	Message Formats	5
1.3	Protocols	7
1.4	Protocol Services	9
2	General Inter-ORB Protocol	10
3	Simple Network Management Protocol	13
4	Common Management Information Protocol	15
5	Hypertext Transport Protocol	18