

---

# L<sup>A</sup>T<sub>E</sub>X@IVT

---

## IVT B<sub>I</sub>B<sub>T</sub>E<sub>X</sub> / L<sup>A</sup>T<sub>E</sub>X Workshop

Michael Balmer

Kirill Müller

Transport Planning and Systems

November 2011

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Preparation</b>	<b>1</b>
2.1	Adobe Reader (00)	1
2.2	Active Perl (01)	2
2.3	Python (01a)	2
2.4	Putty Collection (02)	2
2.4.1	Getting Access to the IVT SVN Server	3
2.5	TortoiseSVN (03)	4
2.5.1	Getting the BibTeX / L <sup>A</sup> T <sub>E</sub> X environment	6
2.5.2	Commands	7
2.6	WinMerge (04)	10
2.7	MiKTeX (05)	10
2.8	T <sub>E</sub> X2html (06)	11
2.9	L <sup>A</sup> T <sub>E</sub> X2rtf (07)	11
2.10	WinEdt (08)	12
2.10.1	Installation and Setup	12
2.10.2	Dictionaries	13
2.10.3	mergeAll.pl	14
2.10.4	Producing a PDF from an Existing Paper in the IVT SVN	15
2.11	Summary	16
<b>3</b>	<b>Using BibTeX Entries</b>	<b>16</b>
3.1	Create the Complete Bib-Entry file (all.bib)	16
3.2	Create a PDF File with all Bib-Entries Available	16
3.3	Create an RTF File	17
3.3.1	LaTeX2RTF	17
3.4	Create an HTML File	17
3.4.1	TTH	17
3.5	Create Your Own Reference List	18
3.5.1	Create Your Personal Working Directory	18
3.5.2	Find the Bibliography KEY	18
3.5.3	Create the PDF and HTML	19
3.5.4	Add your Personal Working Directory to SVN	20
<b>4</b>	<b>Adding BibTeX Entries</b>	<b>20</b>
4.1	Files	20

4.2	File Partitioning . . . . .	22
4.2.1	translations.txt . . . . .	22
4.2.2	main.bib . . . . .	22
4.2.3	The Entry files (ENTRYTYPE.bib) . . . . .	23
4.2.4	Examples . . . . .	25
4.3	Hints for Adding Entries . . . . .	25
<b>5</b>	<b>Writing Papers in L<sub>A</sub>T<sub>E</sub>X Using the Bibliography Environment</b>	<b>27</b>
5.1	Document Layouts . . . . .	27
5.2	Start your own L <sub>A</sub> T <sub>E</sub> X Document . . . . .	27
5.2.1	Papers . . . . .	27
5.2.2	Dissertations . . . . .	30
5.3	How to write L <sub>A</sub> T <sub>E</sub> X . . . . .	30
5.3.1	Excel2LaTeX . . . . .	30
5.4	Counting words . . . . .	30
<b>6</b>	<b>Summary</b>	<b>32</b>

## List of Figures

## List of Tables

# 1 Introduction

This is a small guideline for using Bib<sub>T</sub>E<sub>X</sub> and L<sub>A</sub>T<sub>E</sub>X environment at the IVT. The operating system used here is *Microsoft Windows*. If there is any question or suggestion about this topic, please contact [muelleki@ivt.baug.ethz.ch](mailto:muelleki@ivt.baug.ethz.ch).

We use the following formatting convention throughout this document:

**Paths, URLs, and console commands** are written in a non\_proportional/font

**File formats** are written in non-proportional *italics*.

**User interface** commands are shown like *this*.

**Keyboard** commands are shown like **this**.

**Software packages** are written in *italics*.

**Subversion** commands are written in non-proportional bold.

## 2 Preparation

There are several things to take care of before you can start using the environment. Please follow the instructions below.

Note that under Public\Daten\Install you find the collection of software (in the folder allYouNeedForLaTeX) needed for using all kinds of features. The packages are numbered from **00** to **10**. This reflects the order as described in here. If you want to install the software from there, just copy the whole directory to a local directory on your computer, e.g., My Documents\install.

Note that we will use example paths and names that will be consistent during the whole paper, e.g., the path to the install directory is now [My Documents\install]. The brackets denote that you could change that to another path.

### 2.1 Adobe Reader (00)

In principle you can use any kind of PDF viewer. However, many PDF viewers treat the PDF file as static and are not prepared for the file being replaced or modified in the background, including current versions of *Adobe Reader*. The PDF viewer bundled with T<sub>E</sub>XWorks watches for changes to the PDF file. T<sub>E</sub>XWorks will be installed automatically when installing MiK<sub>T</sub>E<sub>X</sub>, cf. Section 2.7. We will show how to configure our text editor to use T<sub>E</sub>XWorks as PDF viewer.

On the other hand, *Adobe Reader* is the standard reader, and you should always check your documents if they are displayed properly in *Adobe Reader*.

1. Navigate to [My Documents\install]\00\_adobeAcroReader
2. Install AdbeRdr1000\_en\_US.exe

## 2.2 Active Perl (01)

*Perl* is a scripting language. To run *Perl* scripts (files with ending .pl), you need to install its interpreter for *Windows*, provided by <http://www.activestate.com/Products/ActivePerl> (Linux/Unix and Mac already provide the interpreter by default). You need that to use the bibliography environment.

1. Navigate to [My Documents\install]\01\_activePerl
2. Install ActivePerl-5.12.2.1203-MSWin32-x86-294165.msi for a 32 bits system or ActivePerl-5.12.2.1203-MSWin32-x64-294165.msi for a 64 bits system.
3. Define C:\Perl as the destination folder for the program.

After the installation is completed, you can check the functionality by typing

```
perl -v
```

into a *cmd* (command window). A copyright information should appear.

## 2.3 Python (01a)

*Python* is another scripting language. To run *Python* scripts (files with ending .py), you need to install its interpreter for *Windows* (Linux/Unix and Mac already provide the interpreter by default). Scripts that add new entries to the bibliography are written in *Python*.

1. Navigate to [My Documents\install]\01a\_python
2. Install python-2.7.1.msi

After the installation is completed, you can check the functionality by typing

```
python -V
```

into a *cmd* (command window). A copyright information should appear.

## 2.4 Putty Collection (02)

*Putty* is a small collection of executables for *Windows* (<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>) to establish a safe remote connection (ssh) with other computers or servers.

1. Navigate to [My Documents\install]\02\_putty
2. Execute putty-0.60-installer.exe
3. Choose C:\putty as installation directory
4. In C:\putty, make a copy of the executable pscp.exe and rename the copy to scp.exe
5. In C:\putty, make a copy of the executable plink.exe and rename the copy to ssh.exe
6. **Start**→**Control Panel** (if you are working on Windows Vista or Windows 7 ensure that you have a classic view of the control panel)
7. Double-click **System**
8. **Advanced**→Environment Variables
9. In **System variables** double-click on **Path**
10. In **Variable value** add (**do not overwrite!**) C:\putty; at the **very beginning** of the line.  
→**OK**

The last few items add C:\putty to the execution path of *Windows*. With that, you can run those executables in any directory of your machine. To check if the settings are correct, just open a *cmd* (command window) and type in puttygen. If the *PuTTY Key Generator* appears, everything is ok.

#### 2.4.1 Getting Access to the IVT SVN Server

The Bib<sub>T</sub>E<sub>X</sub> / L<sub>A</sub>T<sub>E</sub>X environment is located at the institute server `svn.ivt.ethz.ch`. Therefore, you need access to it. The following *PuTTY* tools are necessary or helpful to get access to the server:

**puttygen** Tool to generate public-private key pairs.

**pageant** Daemon tool for Windows to manage public-private key access to servers.

**putty** SSH client to access servers via the SSH protocol.

First, we need to generate a personal public-private-key pair. This is necessary and useful to get access to the server without typing your server password any time you connect to it.

1. **Start**→**Run...** and type in puttygen
2. check parameters: **SSH-2 RSA** and **2048** bits
3. click on **Generate**
4. move you mouse around until the progress bar reaches the end.
5. at **Key comment** add an appropriate comment (e.g. mylaptop)
6. enter a **passphrase** (a private password) and confirm it
7. **save private key** on your computer (e.g., [D:\whereever\mykey].ppk)
8. keep the *PuTTY Key Generator* open for the next step.

Next, you have to copy the public key to the IVT SVN server.

1. If you are outside the ETH network, start a VPN connection to the ETH network. The details are documented on the ETH web pages at <https://www1.ethz.ch/id/services/list/network/vpn/>.
2. **Start**→**Run...** and type in `plink [username]@svn.ivt.ethz.ch`. Substitute [username] by your n.ethz user name.
3. A password prompt should appear in a new console window. If it does not, contact `support@ivt.baug.ethz.ch` for help.
4. Type in your n.ethz password, confirm with **return**.
5. Upon successful login, a prompt like `[username]@pingelap:~` appears. If it does not, ask `support@ivt.baug.ethz.ch` for access to `svn.ivt.ethz.ch`.
6. In *PuTTYgen*, mark and copy the public key (everything which is shown at the top part of the window).
7. In the *plink* window, type `mkdir .ssh` **return**. Ignore any error messages.
8. Type `cat >> .ssh/authorized_keys` **return**. No prompt should appear.
9. Click the right mouse button inside the *plink* window. The contents you have copied from *PuTTYgen* should be inserted, just as if you had entered it by keyboard.
10. Hit **Ctrl+D**. Now, the prompt should appear again.
11. Optional: Verify that you have created a proper `authorized_keys` file by typing `cat .ssh/authorized_keys` **return**. The contents of your public key should be output.
12. Hit **Ctrl+D**. The *plink* window closes.
13. close *PuTTY Key Generator*.

Now, we are ready to verify the connection to the SVN server.

1. Load your private key in *pageant* by double-clicking `[D:\whereever\mykey].ppk` in *Explorer*
2. When prompted, enter the passphrase for your private key.
3. The daemon (a small computer with a hat) is now shown in the taskbar (bottom right).
4. **Start**→**Run...** and type in `plink [username]@svn.ivt.ethz.ch`
5. A prompt should appear without the need to enter a password first.
6. The connection is working. You can now close that window.

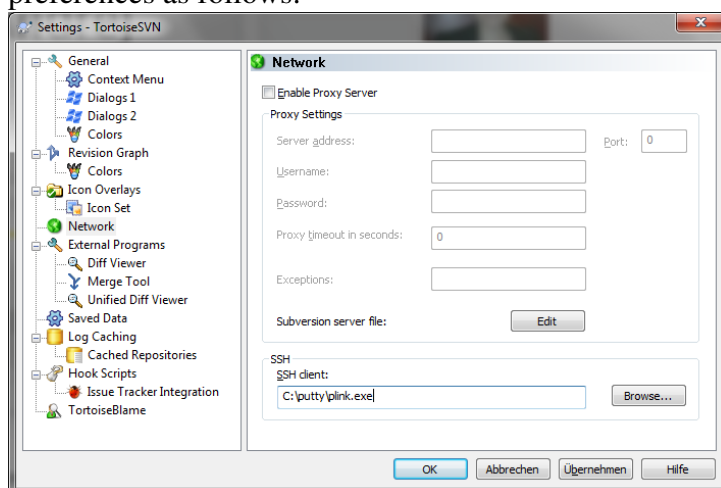
You now have access to the SVN institute server and therefore to the Bib<sub>T</sub>E<sub>X</sub> / L<sub>A</sub>T<sub>E</sub>X environment. Anytime you want to get access to the server, you now just need to load your private key in *pageant* daemon as shown above (if not already running).

## 2.5 TortoiseSVN (03)

This is the SVN client (<http://www.tortoisesvn.org/>) for the SVN repository located on `svn.ivt.ethz.ch`. SVN is a data storage with a version management. In simple words, several

versions of the same file are stored at the same location with the same name. There are features that allows one to get any of those versions of the file. That means it (i) tells you the full history of a file, (ii) it (more or less) guarantees a specific location of that file and (iii) it is also an backup system for your files.

1. Navigate to [My Documents\install]\04\_tortoiseSVN
2. Install *one* of the following files:
  - TortoiseSVN-1.6.12.20536-win32-svn-1.6.15.msi, if you run 32-bit *Windows*
  - TortoiseSVN-1.6.12.20536-x64-svn-1.6.15.msi, if you run 64-bit *Windows*
- Hint: You run a 64-bit *Windows* if there is a folder named Program Files (x86) in the root of your system drive (usually, C:\).
3. Restart *Windows*
4. After reboot, go to **Start**→**Programs**→**TortoiseSVN** →**Preferences/Settings** and set the preferences as follows:



(Consider postponing this setting after trying to checkout something from the IVT SVN: with this setting, an ugly window appears after each operation. SVN might work without this setting; use this setting if you experience connection problems.)

Now, you are able to checkout any part of the SVN repository to your computer. The following subsections gives you a basic introduction how to use svn. But first you need to know some important rules about SVN:

- A checkout produces a **local copy** (called **sandbox**) of the files on your machine. You can work with them even you are not connected. Modifying or deleting files or folders on your machine does not change anything in the repository. You can always get a fresh copy again.
- Differentiate between your sandbox and other files on your machine (which are not part of the repository). We highly suggest to create a special folder in which you check out files

from the repository. Here, we are using [My Documents\sandbox] as the base path to the sandbox.

- **ALWAYS** do an checkout directly **in** the sandbox folder. **NEVER** check out something inside a subdirectory of your sandbox.
- **NEVER** rename files or folders that are part of the local copy of the repositories using *Explorer*. Only rename files or folders via SVN. However, you can safely move or rename those folders that are at the top level of the sandbox.
- **NEVER** create two files or folders with the same name but different case in the same folder hierarchy level (e.g., ivt/doc/papers/STRC and ivt/doc/papers/strc). The (unfortunate) reason for that is that *Windows* is not case-insensitive, but SVN and all other operating systems are. If that happens, some weird conflicts occur (only) on Windows machines.

Because of the above mentioned rules, it makes sense to keep the repository well organized, meaning that folder hierarchies should make sense. (Actually, that always makes sense, even one does not use SVN.)

### 2.5.1 Getting the B<sub>I</sub>T<sub>E</sub>X / L<sup>A</sup>T<sub>E</sub>X environment

We now want to get the B<sub>I</sub>T<sub>E</sub>X / L<sup>A</sup>T<sub>E</sub>X environment with SVN. We suggest to get the complete environment (takes several GB of your hard drive), so that you can see the whole structure. It contains all kinds of papers and presentations produced by IVT members. They do not have to be papers written in L<sup>A</sup>T<sub>E</sub>X, you will also find *Word* documents and *Power Point* presentations.

There are two ways to get the environment: by regular checkout, or by copying and relocating someone else's environment.

**Regular checkout** This is the recommended way if you are connected with a cable to the ETH network (or to a network with a high-speed connection to the ETH). A checkout via WiFi (wireless LAN) or from your home network is also possible but will take several hours.

1. Be sure that your *pageant* daemon is up and running (see above)
2. In *Explorer*, create a folder in which you want to check out parts of the repository (here: [My computer\sandbox])
3. Right-click that folder and choose **SVN Checkout...** and enter the following URL: svn+ssh://[username]@svn.ivt.ethz.ch/var/lib/svn/ivt/doc/trunk
4. Click **OK**

You are now getting the local copy of the complete folder tree of ivt/doc (that will take a while). If the process is interrupted, retry and ignore the warning that the checkout folder is not empty.

**Use someone else's environment** If you can't do or don't want to do a regular checkout but can get a full copy of someone else's SVN environment, you can proceed as follows:

1. Copy someone else's environment to a folder on your hard drive
2. Right-click that folder and choose **TortoiseSVN**→**Relocate...** and enter the following URL: `svn+ssh://[username]@svn.ivt.ethz.ch/var/lib/svn/ivt/doc/trunk`. Usually this means substituting the user name in the existing URL.)
3. Click **OK**

The following two steps are necessary only if you are unsure if the environment you obtained is "clean", i.e., completely in sync with the SVN repository.

4. Right-click that folder again and choose **SVN Update**
5. Right-click that folder again and choose **SVN Revert...** If the list of files is empty, click **Cancel**. Otherwise, check **Select all** and **OK**.
6. Right-click that folder again and choose **SVN Commit...** Check **Show unversioned files**. If the list of files is empty, click **Cancel**. Otherwise, check **Select all**, right-click any file and choose **Delete**. Confirm.

Now your environment is ready to use, just like after a fresh checkout.

## 2.5.2 Commands

Before we focus on L<sub>A</sub>T<sub>E</sub>X, we explain some of the basic functionalities of SVN.

### checkout

**Location** In a NON-SVN folder use right click **SVN Checkout...**

**Purpose** Gets the current version of a folder tree from the SVN repository

**NOTE** For the IVT SVN repository, use the above URL as base. You can browse the SVN repository by pushing the ... button in the checkout dialog.

**NOTE** As long as you only want to use one local sandbox, you need to use checkout only once.

## update

**Location** In a SVN folder use right click *SVN Update*

**Purpose** Gets all the updates of the folder tree below the folder on which you used this command.

**NOTE** You can use this command any time you want. It does NOT destroy anything in your sandbox, it just gives you everything new.

**NOTE** Use is as often as needed.

## add

**Location** In a SVN folder use right click *SVN Add...*

**Purpose** Adds new files and folders to the SVN repository. (It is recursive).

**NOTE** Add new files and folders which you want to store in the SVN repository. Do not add unnecessary files (like temporary files).

**NOTE** Before you add files and folders to the repository, be sure that they are at an appropriate position and have appropriate names.

**NOTE** If you are working on a file, add it to the repository as soon as possible.

**NOTE** To finally put the file into the repository use `commit`

## commit

**Location** In a file in SVN use right click *SVN Commit...*

**Purpose** Puts the changes of the file into the repository.

**NOTE** To check what has been changed in the files, use *SVN Diff*. You can edit the files directly in the *Diff* view.

**NOTE** It is advisable to supply a short log message that describes the nature of the modifications, like “added”, “edited ACME section”, “reorganized paper”, “substituted Creator by Generator”, etc.

**NOTE** If you want to keep all changes, click *OK*.

**NOTE** If you want to restore the file to the previous state, right-click this file (in the *commit* dialog or in *Explorer*) and select *SVN Revert...*

**NOTE** If the file has been modified in the repository since the last update, the commit will fail. In this case, you will have to use the update command first. This may introduce conflicts if the changes overlap in the file, but this will not happen if you are the only one who edits your files.

**NOTE** Try to commit as often as possible, ideally after each “stable” state. By that, you can always easily revert to the last stable state in case your work moves into a wrong direction.

## diff

**Location** In a file in SVN which is modified use right click **SVN Diff**

**Purpose** Shows the changes of your file compared to the version in the SVN repository.

**NOTE** It does only work properly with text files. There is limited support for diffing .doc and .xls files; diffing other kinds of binaries does not work.

**NOTE** It makes sense to check your modifications first before you commit files to the repository.

## remove

**Location** In a file or folder in SVN use right click **SVN→remove**

**Purpose** Removes a file from the repository so that it does not appear anymore when you do a checkout or an update.

**NOTE** After you have used the remove command, you have to use a commit to actually delete the file or folder from the SVN repository.

**NOTE** In fact, the file is not completely removed from the repository. There is always a possibility to get older versions from that file.

## rename

**Location** In a file or folder in SVN use right click **SVN→Rename...**

**Purpose** Moves a file or folder in the repository to a different location, retaining the full history. You can always determine the file's origin by looking at the change log.

**NOTE** After you have used the rename command, you have to use a commit to actually move the file or folder in the SVN repository.

**NOTE** The rename command is implemented as a copy and delete command.

**NOTE** To move a file or folder to a different folder, specify the relative path of the new location, e.g., ../destination\_folder/new\_file\_name.

**NOTE** Moving a big directory tree is faster when using browse (see below)

## copy

**Location** In a file or folder in SVN use right click **SVN→Branch/Tag...**

**Purpose** Copies a file or folder from the repository to a different location, retaining the full history. You can always determine the file's origin by looking at the change log.

**NOTE** The target is a SVN URL, not a working copy path.

**NOTE** After you have used the copy command, you have to use a update command to fetch the copied files from the repository.

## log

**Location** In a file or folder in SVN use right click **SVN Log...**

**Purpose** Lists all committed changes to that file or to all objects in that folder.

## browse

**Location** In a file or folder in SVN use right click **SVN→Repo-browser**

**Purpose** Allows display and modification directly in the repository.

**NOTE** All commands that change objects in the repository ask for a log message.

## ignore

**Location** In a file or folder in SVN use right click **SVN→Add to ignore list→\*.extension**

**Purpose** Stops SVN from offering files for addition.

**NOTE** Files in your sandbox should be (a) added to the repository, (b) ignored, or (c) just not there. This simplifies the definition of a *clean copy*: The sandbox is clean if and only if the commit list is empty (with the **Show unversioned files** checkbox active).

## 2.6 WinMerge (04)

*WinMerge* (<http://winmerge.org/>) is a very nice tool to compare two different text files. It shows the difference between them line by line. The tool does not work for binary files, e.g., *zip* files.

1. Navigate to [My Documents\install]\03\_winMerge
2. install WinMerge-2.10.2-Setup.exe
3. choose **typical installation**, add **Plugins**, leave **Integrate into TortoiseSVN** checked

If you now right-click on a file, you will find **WinMerge** in the context menu.

(Installation of this tool is optional, since *TortoiseSVN* comes with its own file comparison tool.)

## 2.7 MiK<sub>T</sub>E<sub>X</sub> (05)

*MiK<sub>T</sub>E<sub>X</sub>* is a L<sup>A</sup>T<sub>E</sub>X implementation for Windows (<http://miktex.org>). It contains many hundreds of packages. All these together are the basis for compiling L<sup>A</sup>T<sub>E</sub>X documents, such that it produces PDF, PS or DVI files.

1. Navigate to [My Documents\install]\05\_miktex
2. Install basic-miktex-2.9.3972.exe

This installation process may take several minutes. When it is finished, additional command line commands are available, like `pdflatex` or `bibtex` that are necessary to produce PDF documents out of  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  files. Missing  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  packages will be installed as necessary upon user confirmation. (Another option is to get the full installation of Mik $T_{\text{E}}\text{X}$ , which takes more than 1 GB of hard drive space. Run `setup-2.9.3959.exe` and follow the instructions for this option.)

## 2.8 $T_{\text{E}}\text{X}2\text{html}$ (06)

The program  $T_{\text{E}}\text{X}2\text{html}$  creates a HTML page out of a  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  document (<http://hutchinson.belmont.ma.us/tth>). It knows only very basic functionalities and therefore, the resulting HTML file looks a little bit crappy. But it is still useful for a simple conversion.

1. Navigate to [My Documents\install]\06\_tth and unzip `tth_exe.zip`
2. create the directory [C:\tth] (Note: It is highly recommended to install it directly under C: and to use a folder name with no special characters in it, e.g., spaces).
3. copy all unzipped files into [C:\tth]
4. **Start**→**Control Panel** (if you are working on Windows Vista or Windows 7 ensure that you have a classic view of the control panel)
5. Double-click **System**
6. **Advanced**→**Environment Variables**
7. In **System variables** double-click on **Path**
8. In **Variable value** add (**do not overwrite!**) [C:\tth] ; at the **very beginning** of the line.  
→**OK**

The last few items adds [C:\tth] to the execution path of *Windows*. With that, you can run those executables in any directory of your machine. To check if the settings are correct, just open a *cmd* (command window) and type in `tth -help`. A copyright Information should appear.

## 2.9 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2\text{rtf}$ (07)

The program  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2\text{rtf}$  (<http://latex2rtf.sourceforge.net/>) creates an RTF document out of a  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  document. It knows only very basic functionalities and therefore, the resulting RTF file looks a little bit crappy. But it is still useful for a simple conversion.

1. Navigate to [My Documents\install]\07\_latex2rtf

2. install latex2rtf-1.9.19a\_win.exe

Finally, you have installed everything you need to gather, compile, write and commit L<sup>A</sup>T<sub>E</sub>X papers inside the IVT B<sub>I</sub>T<sub>E</sub>X / L<sup>A</sup>T<sub>E</sub>X environment.

## 2.10 WinEdt (08)

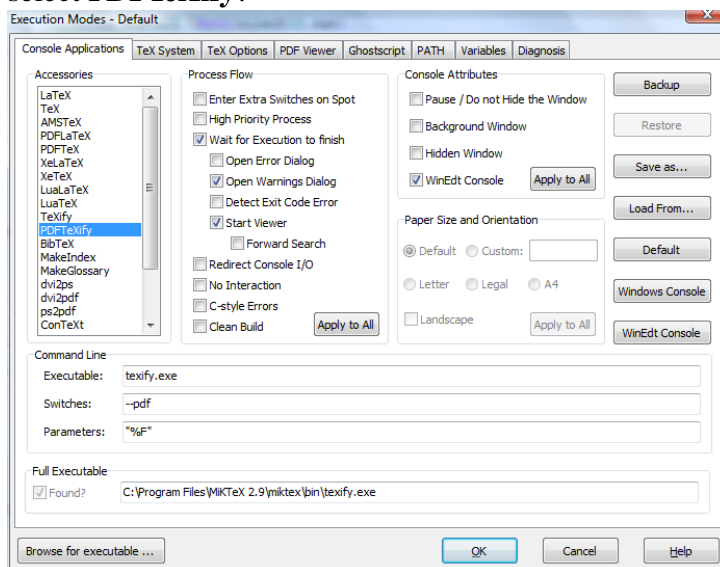
*WinEdt* is a very useful program to write L<sup>A</sup>T<sub>E</sub>X documents (<http://www.winedt.com/>), but it is not necessary. Most colleagues of the IVT that are already work with the environment are using this program already.

### 2.10.1 Installation and Setup

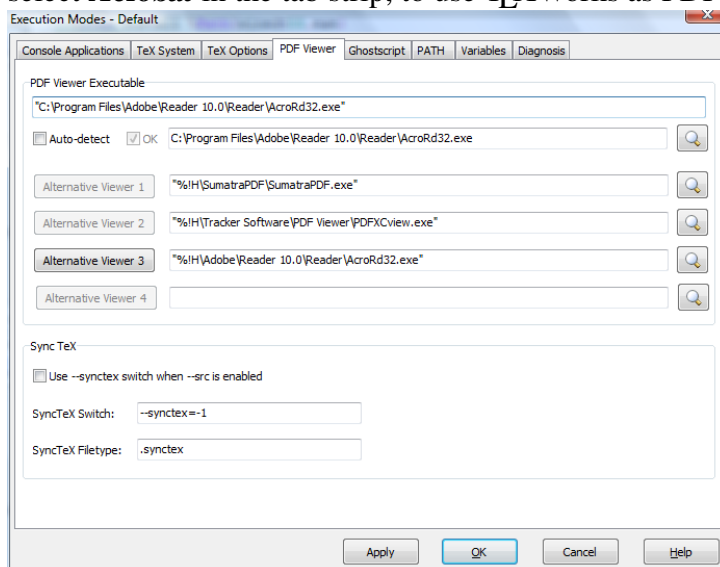
*WinEdt* offers a huge amount of functionalities and different configuration possibilities. The configuration shown here is just a simple one, but it's good enough to work efficiently with *WinEdt* in combination with the IVT B<sub>I</sub>T<sub>E</sub>X / L<sup>A</sup>T<sub>E</sub>X environment. If a configuration is not shown, it is kept to the default.

1. Navigate to [My Documents\install]\08\_winEdt
2. Install winedt60.exe
3. Start *WinEdt*. It shows the *Configuration Wizard* please close it
4. **Options**→**Options Interface...**
5. In the left menu **Formatting: Wrapping, Environments...**→**Wrapping**
6. Overwrite the current file content with the  
[My Documents\install]\08\_winEdt\Wrapping.ini file content..
7. Save the file
8. In the left menu **Backup, Auto Saving, File Status...**→**Backup**
9. Overwrite the current file content with the  
[My Documents\install]\08\_winEdt\Backup.ini file content..
10. Save the file
11. In the left menu **Backup, Auto Saving, File Status...**→**Auto Saving**
12. Overwrite the current file content with the  
[My Documents\install]\08\_winEdt\AutoSave.ini file content..
13. Save the file
14. Click on the **Load Current Script** button (in the up-left corner of the options interface)
15. open the *Execution Modes* via **Options**→**Execution Modes...**. The only L<sup>A</sup>T<sub>E</sub>X compile mode we will use is *PDF TeXify*. Therefore, we configure only this one.
  - a) →**Console Applications**

b) select **PDF $T_{E}X$ ify**:



c) select **Acrobat** in the tab strip, to use  $T_{E}X$ Works as PDF viewer:



## 2.10.2 Dictionaries

That's already enough to work with *WinEdt* nicely. *WinEdt* also provides dictionaries to correct typos. If you want to add and or replace the preset English dictionary with a German or a Swiss-German one, do the following:

1. Unzip the dictionary you like (e.g. `de_neu.dic`). It must be on the  
`[My Documents\install]\09_Dictionaries`  
 folder
2. **Options**→**Options Interface...**
3. In the left menu **Dictionary Manager: Word Lists, Spelling...**→**Word Lists (Dictionaries)**

4. Add the next text to the end of the dictionaries.ini file (ensure to replace the [My Documents\install] folder for your working folder)

```

DICTIONARY="de_new"
FILE="[My Documents\install]\09_Dictionaries\de_neu.dic"
ENABLED=1
MODE_FILTER=""
LOAD_ON_START=0
SAVE_ON_EXIT=0
ADD_NEW_WORDS=0
USE_FOR_COMPLETION=0
ALLOW_COMPOUNDED_WORDS=0

```

5. Save the file

### 2.10.3 mergeAll.pl

Anytime you update/change/add bibliography entries in the environment, you need to run a special *Perl* script that can be found in the SVN at `ivt\doc\trunk\_latexfiles\mergeAll.pl`. Navigate to the folder tree `C:\[pathToYourSandbox]\_latexfiles` and run that script via a double-click. It creates the file `C:\[pathToYourSandbox]\_latexfiles\bibs\all.bib` (and others). To run that script from any folder (and therefore, from any program) on your machine, you need to create with a text editor a *bat* file containing the following lines:

```

C:1
cd "C:\[pathToYourSandbox]\_latexfiles"2
mergeAll.pl3

```

Store that file as `[C:\putty\mergeAll.bat]`. You have already added `[C:\putty]` to the system path. So, you do not need to add another path to the environment variables.

1. **Start**→**Run...** and type in `cmd`
2. in the console, type in `mergeAll` and hit **return**

If many logging information appear ending with:

```

...
done.

```

---

<sup>1</sup>change to that hard drive partition where you keep your *SVN* sandbox

<sup>2</sup>go to folder `_latexfiles`

<sup>3</sup>run the *Perl* script

setting language specific header to ...

done.

done.

then the *Perl* script “mergeAll” can be run everywhere on your folder tree.

Now, if you want to call that program from inside of *WinEdt* you can add another execution button:

1. **Options**→**Options Interface...**
2. In the left menu **Menus and Toolbar**→**Main Menu**
3. Overwrite the current file content with the  
     [My Documents\install]\08\_winEdt\MainMenu.ini file content.
4. Save the file
5. Click on the **Load Current Script** button (in the up-left corner of the options interface)

Now, the mergeAll program can be executed inside *WinEdt* via **Accessories**→**mergeAll**. A command window should appear and producing the logging output as you know from above. Then the window will disappear again.

#### 2.10.4 Producing a PDF from an Existing Paper in the IVT SVN

To test, if everything is working fine, we will compile this working paper which is located in SVN at

C:\[pathToYourSandbox]\papers\workingpapers\2011\bibtex-latex-workshop.

1. **Project**→**Open Project...**
2. Navigate to the directory var\lib\svn\ivt\doc\trunk\papers  
     \workingpapers\2011\bibtex-latex-workshop
3. open bibtex-latex-workshop.prj
4. click on **Erase Working Files** (recycle bin button)
5. run the mergeAll script with **Accessories**→**mergeAll**
6. create the PDF by clicking **PDF Texify** (button with the teddy bear in front of the Acrobat icon)

Then, the PDF will be created and *Acrobat Reader* will show the resulting PDF.

Those three clicks should always be done to create the pdf from a L<sub>A</sub>T<sub>E</sub>X project. You can define keyboard shortcuts for the three clicks as seen above to make the use more comfortable.

## 2.11 Summary

It is not necessary to use / install all of the above mentioned programs and configurations, but it give one a complete overview. If you have questions or problems do not hesitate to ask / write / call [muelleki@ivt.baug.ethz.ch](mailto:muelleki@ivt.baug.ethz.ch).

## 3 Using BibTeX Entries

This section shortly describes how you can gather all references stored in the bibliography database. You do not need to write any L<sup>A</sup>T<sub>E</sub>X or bib-entries, you just need to compile stuff.

### 3.1 Create the Complete Bib-Entry file (all.bib)

If you have done the preparation as shown in Section 2, you have the following folder trees in your sandbox:

- `_latexfiles`
- `misc`
- `etc...`

In `_latexfiles` you find a script called `mergeAll.pl`. It produces the complete list of bib-entries and stores it under `_latexfiles/bibs/all.bib`<sup>4</sup>.

To run that script just double-click it! Anytime you do an SVN update that includes the `_latexfiles` directory you also should run this script.

### 3.2 Create a PDF File with all Bib-Entries Available

In the folder `misc/allbibs/ivt` you find the following files:

- `ivt.tex`: L<sup>A</sup>T<sub>E</sub>X file to create all references in the bib database
- `latex2pdf.bat`: runs the commands to create a PDF
- `latex2html.bat`: runs the commands to create a HTML page
- `cleanLatex.bat`: removes all temporary files that are created while compiling the L<sup>A</sup>T<sub>E</sub>X file.
- `ivt.prj`: WinEdt project file (you can ignore it)

---

<sup>4</sup>It also creates IVT style files

If you double-click `latex2pdf.bat`, an `ivt.pdf` will be created (apart from other – unimportant – files) containing all available bib-entries in the IVT reference style for English papers.

If you double-click on `cleanLatex.bat`, all created files will be removed again.

If you want to create the references for German papers, open the file `ivt.tex` with *WinEdt* and change line number 14 from

```
\newcommand{\myfirstlang}{english}
```

to

```
\newcommand{\myfirstlang}{german}.
```

Save it and compile the file again with `latex2pdf.bat`.

### 3.3 Create an RTF File

The above create the right reference style in PDF format. From that, you can copy and paste a reference in your paper (e.g., a *Word* document). Unfortunately, copying text from a PDF does not preserve text styles like **bold** or *italic*. Therefore, you have to adapt that manually.

#### 3.3.1 LaTeX2RTF

There is also an open-source project, called *LaTeX2RTF*, which can convert a LaTeX document into RTF. You can download and install it from <http://sourceforge.net/projects/latex2rtf/>. The use of that tool will be shown in the workshop. It supports the formatting styles, but unfortunately no UTF-8 encoding. That means that you have all the fancy formatting stuff but, e.g., an “ö” will be encoded very weird (i.e., “ $\tilde{A}f$ ”)... \*gmpf\*.

### 3.4 Create an HTML File

From the above, creating an RTF does not help that much. But you also can create an HTML from the L<sub>A</sub>T<sub>E</sub>X file. With that, copy and paste into a *Word* file is much nicer.

#### 3.4.1 TTH

The open-source tool, called *TTH* (TeXTToHTML), converts L<sub>A</sub>T<sub>E</sub>X into HTML. You can download it from <http://hutchinson.belmont.ma.us/tth/>. The use of that tool will be shown in the workshop. It supports formatting style and knows how to handle umlauts. After generating

the HTML page you can easily copy and paste it into a *Word* document. The text styles are preserved. We suggest to use that mechanism for *Word* files.

In *misc/allbibs/ivt* a BAT file exists (**latex2html.bat**) which creates the HTML using **TTH**.

## 3.5 Create Your Own Reference List

To create your own reference list (e.g., when you have to respect a specific order, like for TRB papers) you have to write a tiny little bit of L<sub>A</sub>T<sub>E</sub>X.

### 3.5.1 Create Your Personal Working Directory

An example how to do that is shown in *misc/unsortbibs/example*. It contains the following files:

- `cleanLatex.bat`
- `example.prj`
- `example.tex`
- `latex2html.bat`
- `latex2pdf.bat`

The functions of the three BAT files are already explained above. This L<sub>A</sub>T<sub>E</sub>X document creates all references in a predefined order. If you now want to have your own reference list created, create a folder with your username (e.g. *misc/unsortbibs/balmermi*), copy all files from *misc/unsortbibs/example* into it, change the name of the L<sub>A</sub>T<sub>E</sub>X file to your name (e.g. *balmermi.tex*) and open it with an editor (*WinEdt*).

The file is well documented, but still here some remarks:

- On line 14 you define if you want to create German or English references.
- Starting at line 72, the references are defined. In the original file, there is only one entry that reads `\nocite{*}`, meaning that all references are to be included. You can replace this entry by several `\nocite{KEY}` entries.

Add your list in the same manner as shown in the example.

### 3.5.2 Find the Bibliography KEY

To find out which *KEY* to use, open the file *doc/\_latexfiles/bibs/all.bib* with your preferred text editor. (If this file does not exist, create it as explained in Section 3.1.) Since the bibliography

files are plain text files, you are able to find the references with just a normal search in your text editor. Therefore, you can search for authors, titles, etc...

Each bibliography entry looks like this:

```
@TYPE{KEY,
  ATTRIBUTE = {VALUE},
  ATTRIBUTE = {VALUE},
  ...
  ATTRIBUTE = {VALUE},
}
```

The KEY is always defined at the first line of an entry. As an example, let us use the entry for Axhausen's Book "Moving Through Nets". So we search for "Axhausen", "moving" etc. and will sooner or later find the following entry:

```
@BOOK{Axhausen_2006,
  EDITOR = Axhausen,
  TITLE = {Moving Through Nets: The ...},
  PUBLISHER = Elsevier,
  YEAR = {2006},
  address = oxford,
}
```

The key for that entry is therefore Axhausen\_2006. Therefore, we will add the following citation in the balmermi.tex file:

```
\nocite{Axhausen_2006}
```

When you are done adding your citations, save and close the file.

### 3.5.3 Create the PDF and HTML

To create the *pdf* (and *html*) out of your *tex* file, you need to edit the two *bat* files latex2pdf.bat and latex2html.bat. Open them with your preferred text editor.

If you have copied them from the example, you will find the word `example` many times. It defines your L<sub>A</sub>T<sub>E</sub>X document which you want to compile. In the example showed above, the L<sub>A</sub>T<sub>E</sub>X document is named *balmermi.tex*. Therefore, replace all occurrences of `example` with `balmermi` (without the ending `.tex`). That's all!

Save and close the two *bat* files and double-click them to create the *pdf* or *html*.

### 3.5.4 Add your Personal Working Directory to SVN

When you are happy with your personal working directory, you should add it to SVN, so that you can reuse it later again. You should add only the folder and files which are necessary. In the above showed example, you should add

- the balmermi folder and the files:
- cleanLatex.bat (BAT to remove unnecessary files)
- balmermi.tex (the L<sup>A</sup>T<sub>E</sub>X file)
- latex2html.bat (BAT to create the HTML)
- latex2pdf.bat (BAT to create the PDF)

All intermediate files (e.g., .aux or .log) should not be added to SVN. Also the PDF and the HTML files do not have to be added, since they can be created again. To stop SVN from showing these files as “non-versioned”, you can ignore them.

To add the folder and the files to SVN, use command `add` (as described in Section 2.5) for the folder and the files. Then use the command `commit` for the files to finally commit them. Your working directory is saved and backed up now.

## 4 Adding BibTeX Entries

Here we explain how to add new bibliography entries to the BibTeX / L<sup>A</sup>T<sub>E</sub>X environment. The system gets more and more attractive when more and more people are using it. The reasons for that are:

- Each entry has to be added only once. You are able to reuse an entry as often you want. Also, other users get access to the entries without rewriting them over and over again.
- You do not need to care about the way a reference will be created. This will be automatically done via the L<sup>A</sup>T<sub>E</sub>X style definitions.
- Entries will be automatically adapted to the German or English version. (E.g., in English references we have to write “Zurich” while in German references the town is written as “Zürich”).

### 4.1 Files

The following files are used to define Bibliography Entries:

**ivt/doc/\_latexfiles/bibs/main.bib** This file contains

- Journal and Magazines,
- Publishers and
- Authors.

**ivt/doc/\_latexfiles/styles/translations.txt** This file defines language specific words, like:

- The months (e.g., March vs. März)
- Addresses (e.g., Zurich vs. Zürich)
- Organizations (e.g., Swiss Federal Roads Authority vs. Bundesamt für Strassen)
- Universities and schools (e.g., ETH Zurich vs. ETH Zürich)
- Departments, institutes and groups (e.g., Institute for Economic Research (IRE) vs. Istituto Ricerche Economiche (IRE))
- Types for technical reports (e.g., Working paper vs. Arbeitsbericht)
- Types for research reports (e.g., Research Report vs. Forschungsbericht)
- Types for unpublished references (e.g., internal presentation vs. Interner Vortrag)
- Miscellaneous publications like webpages or software
- the word “forthcoming” vs. “im Druck”

**ivt/doc/\_latexfiles/bibs/{bib-type}.bib** The actual bibliography entries. The following types are available:

- \_latexfiles/bibs/article.bib
- \_latexfiles/bibs/book.bib
- \_latexfiles/bibs/incollection.bib
- \_latexfiles/bibs/inproceedings.bib
- \_latexfiles/bibs/manual.bib
- \_latexfiles/bibs/mastersthesis.bib
- \_latexfiles/bibs/misc.bib
- \_latexfiles/bibs/phdthesis.bib
- \_latexfiles/bibs/proceedings.bib
- \_latexfiles/bibs/researchreport.bib
- \_latexfiles/bibs/techreport.bib
- \_latexfiles/bibs/unpublished.bib

The file `ivt/doc/_latexfiles/bibs/all.bib` should not be edited since it will be created from the files above, as explained in Section 3.1.

Each of the above mentioned files are simple ASCII files. Therefore you are able to edit them with your preferred editor (e.g., *WinEdt*).

## 4.2 File Partitioning

Since the above mentioned files are simple ASCII documents, it makes sense to define a way how the files should be partitioned such that entries are easier to find.

### 4.2.1 translations.txt

The `translations.txt` file is a simple three-column table separated with **tabs**. It has the following form:

```
KEY [tab] ENGLISH_VALUE [tab] GERMAN_VALUE [return]
```

A line starting with the `%` character is a comment line. This line will be ignored by BibTeX. It is an appropriate way to post comments or suggestions for the other users.

Use an appropriate **KEY**. The rules are:

- only characters small characters and dashes (-). No special characters like umlauts
- add a new triplet such that the keys of each partition are ordered lexicographically.
- **addresses** as simple but unambiguous words
- **organizations** starts with *org-*
- **universities and schools** starts with *uni-* or *school*
- **departments, institutes and groups** starts with the key of the organization, university or school as explained above, followed by a dash and the abbreviation of the group
- **types of technical reports and research reports** starts with *rep-*
- **types of unpublished papers** starts with *unpub-*
- **types of misc** starts with *misc-*

For all the above mentioned groups into the translation file, please sort them **lexicographically** according to the **key**!

### 4.2.2 main.bib

The `main.bib` file is well documented. Adding a new journal, magazine, publisher or author is done according to the following syntax:

```
@string(KEY = "STRING")
```

The *STRING* of an author is always written as:

SURNAME, FIRSTNAME SECONDNAME ETC

Please write down the complete name. If you know only the initials of the *FIRSTNAME*, etc. write them with appending dot (.). There are already many examples in `main.bib`, so just keep to that syntax. Please order your entries **lexicographically** according to the **key**!

Adding new authors or journals has been automated. To add an author, execute the script `add_author.py` in `_latexfiles`. (This script requires Python 2.6 or later.) A console window with instructions appears. Enter the author's names, one per line. Finish by pressing **Ctrl+Z return** twice. The authors are added to the `main.bib`, in accordance to above rules. The key is created automatically. To verify the script's operation, execute `diff` on `main.bib`. Do not forget to `commit` these modifications to the SVN repository. – Adding journals works in an analogous ways. Please contact [muelleki@ivt.baug.ethz.ch](mailto:muelleki@ivt.baug.ethz.ch) for comments and suggestions.

### 4.2.3 The Entry files (ENTRYTYPE.bib)

The most tricky part for adding a new bibliography entry is to find out what kind of entry it is. Here is a short overview of the entry types:

**article.bib** An article from a journal or magazine. Therefore, there **MUST** be a journal or a magazine in `main.bib`, e.g., *Transportation*; *Transportation Research Record*; *Zeitschrift für Verkehrswissenschaft*; etc...

**mastersthesis.bib** A master thesis (or “Diplomarbeit”).

**phdthesis.bib** A PhD thesis (or “Doktorarbeit”, Dissertation).

**techreport.bib** A report published by a school or other institution, organizations or groups, usually numbered within a series. Also semester projects but no master / diploma thesis and no dissertations. Also, do not add research report here!

**researchreport.bib** A report published by a school or other institution, organizations or groups. Compared to `techreport.bib`, research reports **MUST** have a **client**.

**manual.bib** A technical documentation. Typically written by an institution, sometime there is an author. It is **NOT** a TechReport!

**misc.bib** Use that only when nothing else fits. Typical examples are webpages and software.

**unpublished.bib** Everything which is **NOT** going to be published. Typical examples are presentations, internals (presentations, seminars, org-meetings, tech-meetings, etc...), personal discussions, etc. It is **NOT** a paper which is going to be published somewhere, like PhD-Thesis in progress, Tech-Reports, submitted papers of a conference or a journal, etc. Usually, when you write a paper, you do not refer to something that is unpublished. But sometimes, you want to add stuff that you have done (especially when it comes to the “Jahresbericht”).

**book.bib** A book with an explicit publisher. NO Journals like TRR! E.g., Arentze and Timmermans: “ALBATROSS: A Learning-Based Transportation Oriented Simulation”; Bieger, Laesser and Maggi: Jahrbuch 2002/2003 Schweizerische Verkehrswirtschaft; etc.

There are **two types of books**. One has an explicit **author** which actually wrote the whole book (like Ferber’s Book “Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence”). The other has an **editor** which collects different papers from other authors. A known example is Axhausen’s book “Moving Through Nets: The Physical and Social Dimensions of Travel”.

**incollection.bib** A part of a book with its own title. Therefore, the book **MUST** exist in the *book.bib* as a book with an explicit **editor**! NO Article in a Journal or magazine! E.g., Axhausen in Jones (1990); Balmer et al. in Timmermans (2005); Balmer et al. in Bieger et al. (2005); etc.

For example: Axhausen has written the introduction for the book “Moving Through Nets”. If you want to refer to his introduction it is an *incollection* entry. The book itself is part of *book.bib*.

**proceedings.bib** The proceedings of a conference. NO Journals like TRR! The papers can be published as CD-ROM, on the web or similar. If the paper of a conference is published in a book DO NOT put it here. It **MUST** be a **BOOK** entry with the paper as an **INCOLLECTION** entry. Proceedings examples are: STRC (web), IATBR (CD-ROM), TRB annual meeting CD-ROM

**inproceedings.bib** A paper published in a **PROCEEDINGS**. Therefore, the conference proceedings **MUST** exist in the *proceedings.bib*! NO Article in a Journal or magazine! NO published book from a conference! E.g., Axhausen and Goodwin in DRIVE (1991); Balmer et al. in AAMAS (2004)

At the beginning of each of the bib-entry files, a precise description is given, how to add a specific entry. More precisely, each file is separated into several parts:

1. Bib-Entry type description
2. Description of the bib-entry **KEY**
3. Description of the available attributes
4. Example entries
5. TODO entries
6. Actual bib-entries

Please strictly follow the description given for each **BIB** file.

## 4.2.4 Examples

Under `misc/reflists/examples` you will find a L<sub>A</sub>T<sub>E</sub>X document which produces all example references shown in each bib-entry file (`ENTRYTYPE.bib`). This could also help to see how the entries will be used.

## 4.3 Hints for Adding Entries

Here are some hints which probably helps to create an entry easily. It is kind of a “cookbook” which you may follow.

1. Run an update on `_latexfiles` (see Section 2.5) to get the newest version of the BIB files.
2. Run the `mergeAll.pl` script.
3. Before you actually start adding a new entry, be sure that it does not already exists. Check that by compiling the L<sub>A</sub>T<sub>E</sub>X document under `misc/allbibs/ivt` (see also Section 3.2) and searching for it.

Another way is to open the file `_latexfiles/bibs/all.bib` with an appropriate text editor and search in there.

If you found it, your work is done! Just use it as described in Section 3. If it is not there yet, proceed.

4. Find out what kind of bib-entry your reference is. Sometimes it is simple (e.g., STRC or TRR paper), sometimes it is quite tricky. The reason for that is typically that you already have the reference from another paper or similar. Unfortunately, it is not guaranteed that this reference is “correct” (whatever that means...).

We suggest to search via *Google*, *Scholar Google*, *e-collention*, *NEBIS*, etc. for keywords. Usually, you will find several different ways of writing of the same reference. With that information—normally—you are able to find out if it is a paper from a book, a paper in a journal, and so on.

*HINT: Take some time to find out what it is. You typically spend twice the time as if you are not a careful searcher. But, at the end, it will pay off!*

Therefore, when you have found out, which kind of reference that is, open the correspondent bib-entry file.

5. Copy the first given example entry (it contains all attributes available) into the TODO part and start replacing the values of the attributes by the information of your reference.

**Note**, that attributes written in *capitals* are *REQUIRED* attributes, while the others are optional. If possible, always try to use all attributes.

Note, that it is **HIGHLY RECOMMENDED** to use the formats as given by the examples at the top of each bib file. Unfortunately, if you miss one of the brackets, or a comma,

or something else, that looks unimportant, there will be strange error messages while compiling, that makes it very hard to find out what's wrong.

6. Use the *keywords* for authors, publishers, organizations, schools, types, month, address, etc. instead of the names. If an author, publisher, ..., address, etc. is not yet present in the `main.bib` or `translations.txt`, resp., add them there.
7. At last, add the **correct key** to that entry and save the file(s).
8. Run the `mergeAll.pl` script.
9. Compile either your own reference list which includes the new entry (as shown in Section 3.5) or just compile the L<sub>TeX</sub> document under `misc/allbibs/ivt` (see also Section 3.2).
10. If the reference looks alright, move your bib-entry in the BIB file from the TODO items to the place where it belongs to (**keys are sorted lexicographically!**).
11. Save the file and run `mergeAll.pl` again. Now, you can use it as described in Section 3.
12. Do not forget to provide your new entry to the other users. For that, use the `commit` command on the folder `_latexfiles` (see Section 2.5). Be sure that the only files which will be committed to the repository are `main.bib`, `translations.txt` and/or `ENTRYTYPE.bib`.

Add an appropriate log message to your `commit` and click **ok**. Now, all other users can now use this new bib-entry.

If you skip this step, your entry might get lost or corrupted during subsequent update runs.

Surely, this process sounds quite time consuming. But just think about how much time you spend for writing a single reference entry for a working paper. Then add the time you spend for reformatting the entry, because somebody complains about the formatting. Then add the time you spend for reformatting the entry again, because you want to send the paper to a conference with its own reference style. Multiply that time with the number of time you write the same reference again for another paper. Multiply that with the number of persons at the IVT doing the same thing as you for the same reference:

$$T_{\text{BibTeX}} < T_{\text{no BibTeX}} = (t_{\text{add}} + t_{\text{reformat}} + t_{\text{rereformat}}) \cdot p \cdot n$$

We are pretty sure, that the above described cookbook for creating an entry is a little bit faster. By the way, L<sub>TeX</sub> is quite good at typesetting equations, too.

## 5 Writing Papers in L<sub>A</sub>T<sub>E</sub>X Using the Bibliography Environment

Here, you will find a short overview about the different layout types of the Bib<sub>T</sub>E<sub>X</sub> / L<sub>A</sub>T<sub>E</sub>X environment.

### 5.1 Document Layouts

The following document types are provided already:

- Papers:
  - `ivt-wp`: IVT working paper layout for German & English language
  - `ivt-generic`: Generic IVT paper layout for different purposes in German & English language
  - `trb`: TRB conference paper for English language (actually also in German)
  - `isctsc`: International steering committee for travel survey conferences (ask “scandinave” what that is)
  - `heureka`: Heureka Conference (ask “meisterk” or “balmermi” what that is)
- Dissertations:
  - `ivt`: IVT working paper layout for German & English language
  - `eth`: ETH dissertation layout for German & English language (ETH Title page and a fancy document layout made by balmermi)
- CV:
  - a fancy CV layout made by balmermi (for German & English language)
- some miscellaneous reference list documents as already mentioned in the previous sections. (German & English)

It is very easy to change a whole paper from one layout into another. Also the layouts for a dissertation can be interchanged very easily.

If you have another paper layout which should be included into the environment, please contact [muelleki@ivt.baug.ethz.ch](mailto:muelleki@ivt.baug.ethz.ch).

### 5.2 Start your own L<sub>A</sub>T<sub>E</sub>X Document

#### 5.2.1 Papers

In `ivt/doc/papers` all papers should be stored at an appropriate place. We **highly recommend** the following folder structure:

```
ivt/doc/papers/[CONFERENCE_NAME]/[YEAR]/
[A_MEANINGFUL_NAME_OF_THE_PAPER]/[PAPERNAME].tex
```

Please do not use spaces in the file names.

Under papers/exampleConference/2007/example you find an example paper which you can use as a template for your own paper. As an example, let us say, we want to write a paper for *STRC 2011* about *social networks*. Then do the following:

1. in papers, create a folder strc (if not already exists).
2. in papers/strc, create a folder 2011 (if not already exists).
3. add and commit these newly created folders to the SVN repository.
4. In *Explorer*, navigate to papers\exampleConference\2007.
5. Use the branch command to copy example: Right-click the example folder and select **SVN**→**Branch/tag...**. Enter svn+ssh://[username]@svn.ivt.ethz.ch/var/lib/svn/ivt/doc/trunk/papers/strc/2011/socialNetworks as target URL. Do not forget to provide a log message. (You can also perform this operation in the repository browser.)
6. update the folder papers/strc/2011 to load the newly copied files into your working copy.
7. rename example.tex to socialNetworks.tex (using the **SVN**→**Rename...** command).
8. replace the occurrences of example in latex2pdf.bat to socialNetworks (as explained in Section 3.5.3).
9. commit the changes to SVN

Now, you can start editing your own L<sup>A</sup>T<sub>E</sub>X paper. For this, open socialNetworks.tex with your preferred editor.

At the top of the file you are able to define in which language you want to write your paper. Also, the paper layout can be defined here.

In the first part of the document, you will find many different *keywords* to define. Each of them is well-documented. The structure of a keyword definition is:

```
\newcommand{keyword}{value}
```

Please do NOT remove any of the keyword definitions. If you do not want to specify a specific keyword, just leave the value empty. The following keywords are available for any kind of paper layout:

- \mypath

- `\myfirstlang`
- `\mytitlefigure`
- `\mytitle`
- `\myinstitution{EN|DE}`
- `\mynumber`
- `\myyear`
- `\mymonth`
- `\myday`
- `\mywordcount`
- `\mykeywords{EN|DE}`
- `\my{first|second|third|forth|fifth|sixth}author`
- `\my{first|second|third|forth|fifth|sixth}authorREF`
- `\my{first|second|third|forth|fifth|sixth}address`

The last keyword definition is a little bit more complex. Just insert the information according to the example:

```
\newcommand{\myfirstaddress}{  
  \createcontact{\myfirstauthor}%  
  {EDIT_ONLY_HERE}% address line 1  
  {EDIT_ONLY_HERE}% address line 2  
  {EDIT_ONLY_HERE}% address line 3  
  {EDIT_ONLY_HERE}% phone  
  {EDIT_ONLY_HERE}% fax  
  {EDIT_ONLY_HERE}% email  
}
```

The actual document starts with the command `\begin{document}`. It starts with the order of your document heading pages (title page, table of contents, abstracts, etc...). After that, you can add your sections, followed by the bibliography and an appendix. The order in which the different parts appear defines the order in which they will show up in the paper. If you do not want to use a specific part (e.g., a table of content) just comment out the corresponding command.

If you want to split up parts into several files (for better organization, e.g., one file for each section), you can always use the `\input{FILENAME}` command. When compiling, L<sub>A</sub>T<sub>E</sub>X will replace this command with the contents of the given file.

## 5.2.2 Dissertations

The use of dissertation layout is similar to the use of the paper layouts. All dissertations should be stored at an appropriate place under `ivt/doc/dissertations`. We highly recommend the following folder structure:

```
dissertations/[YOUR_ETH_USERNAME]/[DISS_NAME].tex
```

Please do not use spaces in the names.

Under `ivt/doc/dissertations/balmermi` you find an example dissertation (well, actually it's balmermi's dissertation). Follow the same instruction as shown in Section 5.2.1. Apart from the same keywords already mentioned in Section 5.2.1, following additional keywords have to be defined:

- `\myfirstauthortitle`
- `\myfirstauthordoctortitle`
- `\myfirstauthorbirthday`
- `\myfirstauthorcity`

The remaining authors are used to add the examiners.

## 5.3 How to write $L_{A}T_{E}X$

Since  $L_{A}T_{E}X$  is a descriptive way to write a paper, some commands are used to define a section, or a list of items. A nice overview of these commands are shown in the  $L_{A}T_{E}X$  paper `papers/exampleConference/2007/example`. More details will be shown in the workshop.

### 5.3.1 Excel2LaTeX

Since it is cumbersome to write and edit a table in  $L_{A}T_{E}X$ , a nice tool can be used to convert Excel tables into  $L_{A}T_{E}X$ . You can download it from <http://www.ctan.org/tex-archive/support/excel2latex/>. In the workshop, it will be shown how to use it.

## 5.4 Counting words

To count the words in a  $L_{A}T_{E}X$  document, it has to be compiled using `pdflatex` in a special way. There is a script that automates this process and even creates a `tex` file for inclusion into your document. It requires *Python*, version 2.6 or later.

- Compile your document as shown in Section 3.5.3
- From your working directory, call the script `wordcount.py` in the `_latexfiles` directory. For papers/exampleConference/2007/example, you would run  

```
python ../../../../_latexfiles/wordcount.py example.tex
```

from a *cmd* window.
- This generates a file named `mywordcount.tex` in your working directory. You can add this to SVN and `\input` it from your L<sub>A</sub>T<sub>E</sub>X document. (The `trb.tex` template already does this for you – see below.)

For convenience, you can also create a *cmd* or *bat* file or a Windows shortcut (*lnk*).

TRB has a limit on the number of words. In addition, each figure and table counts 250 words.<sup>5</sup> There is some logic in the new `trb.tex` template (the one in `_latexfiles/_layouts`) that saves you from computing the “word equivalents”. For existing papers, just remove any definition of `\mywordcount` from your document; then, the one provided by the template is used.

There are two ways to tweak this:

**Exclude parts of text** The L<sub>A</sub>T<sub>E</sub>X macro `\wcFileName` is defined during counting only. In case you want to exclude some parts, say, the abstract and the bibliography, from the word count, use the following construct in your document:

```
\ifdefined\wcFileName%
\renewcommand{\createabstract}[1]{}%
\renewcommand{\bibliography}[1]{}%
\fi%
```

**Turn off the feature** Redefine the command `\mywordcount`:

```
\renewcommand{\mywordcount}{X words + Y figures + Z tables}
```

Here are some details on the counting algorithm:

- Hyphenated words like “within-day” count as one (just like *Word* counts them)
- Formulae (both inline and separate) count as one word each
- Literature references count as one word
- Text inside figures and tables is not counted
- The title page and header/footer are not counted

The algorithm aims at replicating *Word*’s behavior or even underestimating the word count.

---

<sup>5</sup>A vast underestimation, given that usually “a picture is worth a thousand words”.

## 6 Summary

This document presents a short overview on how to use the IVT Bib<sub>T</sub>E<sub>X</sub> / L<sub>T</sub>E<sub>X</sub> environment. For any suggestions, additional wishes on layouts, and bug reports please contact [muelleki@ivt.baug.ethz.ch](mailto:muelleki@ivt.baug.ethz.ch).