

Using Hebrew LaTeX to create DVI/PS/PDF
documents with hyper-links

Tomer Kol
EE Dept.
Technion

May 5, 1999

Contents

1	Introduction	3
1.1	Background and acknowledgments	3
1.2	General description	3
1.3	How to avoid this document	4
1.4	Recommended reading	4
1.5	Organization of this document	5
2	Hebrew LaTeX	6
2.1	Running \LaTeX	6
2.2	Hebrew package usage	6
2.3	Input code page	7
2.4	Font encodings	7
2.5	Bi-directional sectioning commands	8
2.6	Table of contents, List of figures, List of tables	8
2.7	Defining new theorem like environment	9
2.8	Creating an index	9
2.9	Including eps files	10
2.10	Hebrew calendar	10
2.11	Document class for preparing Hebrew/English or English/Hebrew thesis	11
2.12	Compatibility with LaTeX 2.09 style (deprecated)	13
3	Notes on using vim	15
3.1	Introduction	15
3.2	Fonts	15
3.2.1	Under X11	16
3.2.2	Under Windows	16
3.2.3	Problems with Hebrew fonts on Xwin-32	16
3.3	Entering English and Hebrew	16
3.3.1	Switching between English and Hebrew	16
3.3.2	Hebrew in the command line	18
3.4	The \LaTeX menu	18
3.4.1	NOTES	19
3.4.2	Errors while \LaTeX ing the file	19
3.5	Sample macros and key bindings	20
3.5.1	Default \LaTeX bindings	21
3.5.2	Miscelenous useful options (not related to \TeX)	21

4	Notes on creating DVI/PS/PDF files	23
4.1	Hyperlinks – the <i>hyperref</i> package.	23
4.2	Converting an existing L ^A T _E X document	23
4.3	Putting additional document info (For Acrobat reader)	23
4.4	The <i>showkeys</i> package	24
4.5	The <i>pslatex</i> package	24
4.6	Using Type 1 version of the Computer Modern fonts	24
4.7	Hebrew in the bookmarks sidebar	24
5	Miscellaneous Problems and FAQ	26
A	Sample .vimrc file	27
B	Sample LaTeX files	32
B.1	Sample English only document.	32
B.2	Sample Hebrew/English document.	33

Chapter 1

Introduction

1.1 Background and acknowledgments

This document describes a packaging of $\text{T}_{\text{E}}\text{X}$ related programs and resources, adjusted for using $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ to produce Postscript and PDF documents, with full Hebrew capabilities¹.

This project was supported by funds from the Faculty of Electrical Engineering and the “Information Technologies for Technion Teaching” (ITTT) Project, Technion — Israel Institute of Technology.

Concept:

Tomer Kol, Adam Shwartz, Electrical Engineering Department, Technion.

Design and implementation:

Tomer Kol.

Support:(NOTE: support will start once released).

Irena Abramovitch, Computer Center, Technion.

1.2 General description

For the basic use, the user simply crates an ordinary $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ file, adding a few packages, and the result can be transformed into postscript or PDF files.

The text editor GVIM is used to created the text file *.tex, and serves as an environment² to call other programs using pull-down menus or key sequences. There are several supported features such as, for example, *Make DVI*, which calls $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, makeindex, BibTeX and Hebrew support scripts enough times to resolve references etc. A main advantage of GVIM, beside being a powerful editor, is that it exists with an identical interface both for Unix and PCs running MS windows (95/NT).

The tex file is then converted to DVI using elatex ($\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ based on $\epsilon\text{-T}_{\text{E}}\text{X}$, which supports Hebrew). The DVI file can have hyperlinks usable when viewed with a DVI viewer, such as xdvi, or tags, that after creating a postscript file (using dvips) and then a PDF document (using Adobe Acrobat Distiller or the free ps2pdf utility) will result in a hyperlinked PDF document.

¹The current “official” version of Hebrew support for $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ was created by Boris Lavva

²All the normal command line commands are available (VIM just serves as a front end), so you can use them from a command line or from other editors such as emacs.

1.3 How to avoid this document

You are advised to read this document. However, for the impatient:

1. Open *gvim*
 - Unix: simply type `gvim` from any shell.
 - Win: open a DOS shell and type `gvim`. Optionally create a shortcut icon (right mouse click on desktop, choose “new” then “shortcut” and put `gvim` as the name. Afterwards open the properties dialog — by clicking right mouse button on the icon — and set, in the shortcut page, the directory you would like `Gvim` to start in).
2. Open an existing \LaTeX document using the “File” menu, or create a new one from a template. You can also create a skeleton document or using a wizard, through the “ \LaTeX ” menu (bottom section).

1.4 Recommended reading

NOTE: most electronic documents mentioned are available in common \TeX distribution under the *texmf* directory³. Files with the **dvi** extension can be view by *xdvi* (on Unix) or *windvi* (on Win*).

\LaTeX 2 ϵ in general :

BOOKS :

1. “ \LaTeX : a document preparation preparation system” by Leslie Lamport. second edition Addison-Wesley 1994
2. “The \LaTeX Companion” by Michel Goossens, Frank Mittelbach and Alexander Samarin. Addison-Wesley 1994.

Electronic documents :

1. “The not so short introduction to \LaTeX 2 ϵ .” by Tobias Oetiker et. al. (`texmf/doc/latex/genera`)
2. Short users guide (`texmf/doc/latex/base/usrguide.dvi`)
3. Hypertext reference to \LaTeX 2 ϵ commands.
(`texmf/doc/latex/latex2e-html/index.html`)

Web resources :

1. The [TeX user's group](http://www.tug.org/) (`http://www.tug.org/`)
2. Cambridge university's [Text Processing using LaTeX](http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/)
(`http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/`)
3. The [LaTeX Navigator](http://www.loria.fr/services/tex/english/index.html)
(`http://www.loria.fr/services/tex/english/index.html`)

The graphics package Refer to the *epslatex.ps* file (it's included in our bundle under `texmf/doc/latex/graphics`).

³If you use our Win* distribution the default is `C:\TK\teTeX09\texmf`. On Unix consult your system administrator, or use the “locate” command.

The hyperref package Refer to the *manual.pdf* file (look for it under `texmf/doc/latex/hyperref`).

Other packages Look at the various directories under `texmf/doc`.

1.5 Organization of this document

The rest of this document is organized as follows: Chapter 2 describes the use of Hebrew \LaTeX 2 ϵ , but includes some information (e.g., sections 2.8 and 2.9) that are relevant to \LaTeX in general. Chapter 3 describes the use of the *VIM* editor, focusing on its use as an environment for the creation of \LaTeX documents. Chapter 4 covers topics related to the creation of hyper-linked documents (e.g. PDF) using \LaTeX . Chapter 5 describes miscellaneous problems and FAQ, and will probably expand in future versions of this document.

Appendix A contains a sample *.vimrc* file⁴. Appendix B contains two sample \LaTeX templates, that are also available in our distribution.

⁴On Win* operating systems it is often called *_vimrc*.

Chapter 2

Hebrew LaTeX

The current version of Hebrew support for LaTeX was created by Boris Lavva¹, and is available from <http://www.dsg.technion.ac.il/heblatex/>.

The version described in this manual (and distributed by us) is based on Boris's March 98 release, with additional fixes and adjustments included in the *TKfixes* package. This section is mostly (except for sections 2.1, 2.7, 2.8 and 2.9, which are new) based on the README file from the site mentioned above.

2.1 Running L^AT_EX

- In order to use Hebrew you have to use ϵ -L^AT_EX instead of ordinary L^AT_EX (simply use `elatex foo.tex`).
- Use the `hebrew` argument for the *babel* package (see section 2.2 for details).
- Use the *TKfixes* package to fix various bugs and make necessary adjustments.
NOTE: the *TKfixes* package should be the last package loaded!
- If you want to include an index in your document refer to section 2.8.
- If you want to use L^AT_EX to create PDF documents, refer to sections 4.1 and 4
- If you intend to define new theorem like environments in a Hebrew section of the document refer to section 2.7.
- If you want to include *eps* figures in your document refer to section 2.9.

2.2 Hebrew package usage

The usage is simple, for example:

```
\documentclass[12pt,twoside]{book}
\usepackage[hebrew,english,russian,danish,greek,...]{babel}
```

The default document language is the last specified.

LaTeX 2.09 style is also supported (in compatibility mode) to process existing documents, for example:

¹lavva@tx.technion.ac.il

```
\documentstyle[12pt,hebrew_newcode]{article}
```

NOTE: The use of L^AT_EX 2.09 is **deprecated** (not just for hebrew but for English as well). Use L^AT_EX 2_ε.

Languages can be switched by the following macros:

1. `\sethebrew` or `\unsethebrew`
2. `\selectlanguage{hebrew}`, `\selectlanguage{english}`, etc.
3. `\begin{otherlanguage}{hebrew} ... \end{otherlanguage}`, or any other language
4. `\R{hebrew text}` inside the latin text
5. `\L{latin text}` inside the hebrew text

The first three commands are equivalent: the first one provides switching interface, compatible with L^AT_EX 2.09 hebrew style, while the second and third --- are babel-based interface.

The last two commands are used for language and direction switching inside a paragraph.

`hmbx` and `embox` macros should be used for inserting Hebrew and English text (respectively) inside a mathematical formula.

The `\mbox` (and `\text`, if you're using the *amslatex* package) can be used to enter text using the default outside the mathematical formula. If you use these you will have to use `\R` or `\L` to enter the other language.

2.3 Input code page

Hebrew package supports the following input code pages:

1. 7-bit hebrew encoding, also known as old code, defined by Israeli Standard 960 (`si960`)
2. IBM PC code page also known as pc code (`cp862`)
3. WINDOWS code page also known as new code (`cp1255`)
4. ISO 8859-8 Hebrew/Latin code page commonly used in UNIX (8859-8 is also known as new code)

The default input encoding can be set using option defined above in parentheses with `inputenc` style (but **before** babel), for example :

```
\usepackage[cp1255]{inputenc}
```

Input encoding can be changed also inside the document by the

```
\inputencoding
```

 command, for example:

```
\inputencoding{cp862}
```

If no `inputenc` line is given, the 8859-8 will be used as the default input encoding.

The default (ISO 8859-8) is suitable for both Unix and Windows environments.

2.4 Font encodings

Various font encodings can be used, including, of course, Local Hebrew Encoding (LHE), with the aid of `fontenc` style (but **before** babel), for example:

`\usepackage[LHE,OT2,LCY,T1]{fontenc}`

If no `fontenc` line is given, the LHE will be used as the default hebrew font encoding and OT1 will be used as the default latin font encoding. Many useful font-switching commands are provided in `hebfont` package which can be included *after* the `babel`:

`\usepackage{hebfont}`

These commands are in format `\text{Hebrew text}`, for example: `\textoj{Text printed with Old Jaffa font}`, `\textredis{Text printed with redis font}`.

In addition, for backward compatibility in LaTeX 2.09 documents, the four old font-switching commands are supported:

`{\jm ...}`, `{\oj ...}`, `{\ds ...}` and `{\ta ...}` for Jerusalem, Old Jaffa, Dead Sea and Tel-Aviv fonts respectively.

The following TeX fonts are supported:

- Jerusalem (used as default font) דוגמה
- Dead Sea (used as bold font) דוגמה
- Tel-Aviv (used as sans-serif or typewriter font) דוגמה
- Old Jaffa (used as italic or slanted font and for emphasize) דוגמה
- Hebrew classic normal and slanted (command `\textclas`) דוגמה
- Shalom Old Style (command `\textshold`) דוגמה
- Shalom Script (command `\textshscr`) דוגמה
- Shalom Stick (command `\textshstk`) דוגמה
- Carmel normal and slanted (command `\textcrml`) דוגמה
- Frank Ruehl normal, bold and slanted (command `\textfr`) דוגמה
- Redis normal, bold and slanted (command `\textredis`) דוגמה

2.5 Bi-directional sectioning commands

These commands insert both Hebrew title to the RL table of contents or list of figures/tables, and English (or other LR language) title to the LR table of contents or list of figures/tables.

1. `\bpart{hebrew}{english}`
2. `\bchapter{hebrew}{english}`
3. `\bsection{hebrew}{english}`
4. `\bsubsection{hebrew}{english}`
5. `\bsubsection{hebrew}{english}`
6. `\bcaption{hebrew}{english}`
7. `\bchapternn{hebrew}{english}` - replaces `\chapter*`
8. `\bsectionnn{hebrew}{english}` - replaces `\section*`

A number of additional bi-directional commands are defined in the `hebtech.cls` - thesis document class file, described below.

2.6 Table of contents, List of figures, List of tables

Both Right-to-Left and Left-to-Right versions of the tables (`toc`, `lof` and `lot`) can be generated simultaneously. The LR files have the

default extensions (.toc, .lof and .lot), while the RL files have reversed extensions (.cot, .fol and .tol). For example, to include RL tables at the beginning and LR tables before the end of the document:

```

-----
\begin{document}
\sethebrew
\tableofcontents % Generates .cot file
\listoffigures  % Generates .fol file
\listoftables   % Generates .tol file
...
\unsethebrew
\tableofcontents % Generates .toc file
\listoffigures  % Generates .lof file
\listoftables   % Generates .lot file
\end{document}
-----

```

All 6 files are generated simultaneously in two latex passes !

2.7 Defining new theorem like environment

When you use the `\newtheorem` command you have to spell the name of the environment using tokens like `\alef` and `\bet` as in (note the `\R{}`):

```
\newtheorem{theorem}{\R{\mem\shin\pe\tet}}
```

Table 2.1 lists all the Hebrew letters tokens.

<code>\alef</code>	<code>\zayin</code>	<code>\lamed</code>	<code>\ayin</code>	<code>\resh</code>
<code>\bet</code>	<code>\het</code>	<code>\finalmem</code>	<code>\finalpe</code>	<code>\shin</code>
<code>\gimel</code>	<code>\tet</code>	<code>\mem</code>	<code>\pe</code>	<code>\tav</code>
<code>\dalet</code>	<code>\yod</code>	<code>\finalnun</code>	<code>\finaltsadi</code>	
<code>\he</code>	<code>\finalkaf</code>	<code>\nun</code>	<code>\tsadi</code>	
<code>\vav</code>	<code>\kaf</code>	<code>\samekh</code>	<code>\qof</code>	

Table 2.1: Hebrew letters token names

2.8 Creating an index

In order to create an index use the *makeidx* package. Put in the preamble:

```
\usepackage{makeidx}
```

You also have to put a `\printindex` command where you want the index to appear.

Whenever you want to add a word/term to the index use the `\index` command. Note that what's inside the braces will not show on the text, so commonly you'll repeat it outside, as in:

```
I want this term\index{term} to appear in the index
```

L^AT_EX lists your entries in an auxiliary file **texfilename.idx**. To create the index itself you have to run the *makeindex* program with **texfilename.idx** as an argument, and rerun L^AT_EX.

2.9 Including eps files

The recommended way to include figures is by using *eps* (encapsulated postscript) files, using the *graphics* or the *graphicx* packages (the latter has a slightly improved interface).

You can use various commercial and freeware programs to produce *eps* figures or convert from other formats. Examples of freeware drawing/conversion programs are *Xfig* and *XV* on Unix, and *TKPAINT* (the TK here comes from Tcl/TK) and *ImageMagic* for Win32.

To use *graphicx* put

```
\usepackage{graphicx}
```

And include the *eps* file by the `\includegraphics` command. To get correct alignment it should appear in an English environment. Thus to include an *eps* file in an Hebrew section you should use:

```
\selectlanguage{english}
\includegraphics{picture.eps}
\selectlanguage{hebrew}
```

If you use *graphicx* you can scale your figure in various ways, e.g.:

```
\includegraphics{file.eps} % include figure in the original size
\includegraphics[height=3cm]{file.eps} % scale figure so the hight is 3cm
\includegraphics[width=3cm]{file.eps} % scale figure so the width is 3cm
\includegraphics[scale=1.5]{file.eps} % enlarge figure by 50%
```

The figure can also be rotated, trimmed etc., for further information look at the *epslatex.ps* file (it's included in our bundle under `texmf/doc/latex/graphics`).

An example of a figure environment:

```
\begin{figure}[ht]
  \selectlanguage{english}
  \begin{center}
    \includegraphics{file1.eps}
  \end{center}
  \selectlanguage{hebrew}
  \caption{%
    תרנוכ %
  }
  \label{fig:first1}
\end{figure}
```

2.10 Hebrew calendar

Original Hebrew calendar style was created by Michail Rozman, and corrected by Rama Porrat, Michail Rozman and Dan Haran.

This style was adjusted to L^AT_EX 2e by Boris Lavva, and provided in *hebc* package. It can be included *after* the *babel*:

```
\usepackage{hebc}
```

The following 2 user-level commands are provided with the package:

```
\newcount\hd \newcount\hm \newcount\hy  
\hd=10 \hm=3 \hy=1992  
\HebrewDate{\hd}{\hm}{\hy}
```

which converts the requested Gregorian date to Hebrew calendar date and prints it in Hebrew or English (depends on the current language) and:

```
\HebrewToday or \hebrewtoday  
which prints the current Hebrew date (today) .
```

2.11 Document class for preparing Hebrew/English or English/Hebrew thesis

A useful document class file `hebtech` for preparing M.Sc. and Ph.D. theses for the Technion (English/Hebrew and Hebrew/English) is included with this package too. I believe that it can be adjusted for the needs of other academic institutions in Israel too. This class automatically includes `babel` with `hebrew` and `english` option, `hebfont` and `hebcald` packages.

The original version for LaTeX 2.09 and old code Hebrew was created in 1994 by Irina Abramovici, at Taub Computer Center, Technion.

How to use this document class?

```
\documentclass[12pt]{hebtech}
```

The following parameters should be included in thesis preamble (before `\begin{document}`):

1. Thesis title
`\title{hebrew}{english}`
2. Thesis author
`\author{hebrew}{english}`
3. Department name, e.g. Electrical Engineering Department
`\dept{hebrew}{english}`
4. Domain of research, e.g. Food Processing
`\dom{hebrew}{english}`
5. Main advisor
`\advisorI{hebrew}{english}`
6. Second advisor, if exists
`\advisorII{hebrew}{english}`
7. Main sponsor
`\donorI{hebrew}{english}`
8. Additional sponsor, if exists
`\donorII{hebrew}{english}`
9. Set the `thestyle` counter to:
 - o 1 --- for a research thesis
 - o 2 --- for a project thesis
 - o 3 --- for a final paper`\setcounter{thestyle}{1}`

10. Set grade counter to:
 - o 1 --- for master of science
 - o 2 --- for master
 - o 3 --- for master of technology
 - o 4 --- for doctor of science`\setcounter{grade}{1}`
11. Date is optional, if not included, today's date will be used. The Gregorian date entered will be printed and get converted the Jewish date.
`\date{28}{2}{1998}`
12. Page style is optional too, if not included, plain will be used.
`\pagestyle{thesis}`

In the main document the following additional commands and environments can be used:

1. `\makecover` - creates Hebrew or English cover page, depending on the current language.
2. `\maketitle` - creates Hebrew or English title page, depending on the current language.
3. `\begin{acknowledgment} ... \end{acknowledgment}` - builds the acknowledgment page and adds inserted customized acknowledgment in Hebrew or English, depending on the current language.
4. `\begin{dedication} ... \end{dedication}` - builds the dedication page in Hebrew or English, depending on the current language.
5. `\tableofcontents`, `\listoffigures`, `\listoftables` - builds tables depending on the current language (all 6 tables can be used simultaneously).
6. `\begin{abstract} ... \end{abstract}` - builds the abstract pages in Hebrew or English, depending on the current language, adds contents line to the 2 tables of contents.
7. `\frontmatter`, `\mainmatter`, `\backmatter` - should be inserted at the beginning, before the main part, and at the ending of the thesis to change numbering and other style parameters.
8. `\bibliography{...}` - now adds "References" line to the 2 tables of contents: English and Hebrew.

Some useful bi-directional commands, for figures and tables provided in this file too:

1. `\bcaptionf{hebrew}{english}` - similar to `\bcaption` but answers to thesis formatting requirements at the Technion, used for figures.
2. `\bcaptionff{to RL lof}{to LR lof}{hebrew}{english}` - a customized one for figures.
3. `\bcaptiof{hebrew}{english}` - similar to `\bcaption` but answers to thesis formatting requirements at the Technion, used for tables.
4. `\bcaptiontt{to RL lot}{to LR lot}{hebrew}{english}` - a customized one for tables.

2.12 Compatibility with LaTeX 2.09 style (deprecated)

Generally, the new package is compatible with LaTeX 2.09 style file `hebrew.sty`, i.e. in most cases one can use the new package on old documents without any errors. However, it does not support Plain TeX, please use `heb_macros.tex`, or other TeX macro file for Hebrew instead. A few incompatibilities are described below:

- * The macros with Hebrew names are not provided, for example, macro `\hadgash` when the command name "hadgash" is given in Hebrew is not supported. The reason is that in new LaTeX 2e it is forbidden to use 8-bit macros, because all characters 128-255 became "active".

However, the following English-coded equivalents are supported for backward compatibility with LaTeX 2.09 style and can be used even in the new documents:

1. `\hebday` (replaced "hayom").
 2. `\hebcopy` (for use in letter style)
 3. `\hebincl` (for use in letter style)
 4. `\hebpage` (for use in letter style)
 5. `\hebto` (for use in letter style)
 6. `\hadgash` (produces ``poor man's bold'' - heavy printout, it is advisable to use real bold fonts, e.g. DeadSea instead)
 7. `\piska` (can be used only in compatibility mode)
 8. `\piskapiska` (can be used only in compatibility mode)
 9. `\makafgadol` (`=\textendash`)
 10. `\makafanak` (`=\textemdash`)
 11. `\geresh` (`=\textquoteright`)
 12. `\opengeresh` (`=\textquoteright`)
 13. `\closegeresh` (`=\textquoteleft`)
 14. `\openquote` (`=\textquotedblright`)
 15. `\closequote` (`=\textquotedblleft`)
 16. `\leftquotation` (`=\textquotedblright`)
 17. `\rightquotation` (`=\textquotedblleft`)
 18. `\undertext` (`=\underline`, can be used only in compatibility mode)
- * In `\documentstyle` line the following 3 styles can be used in Hebrew documents:
 1. `\documentstyle[hebrew_oldcode,...]{...}` in old 7-bit documents (instead of "hebrew!").
 2. `\documentstyle[hebrew_newcode,...]{...}` in 8-bit UNIX or Windows documents.
 3. `\documentstyle[hebrew_p,...]{...}` in 8-bit DOS documents.
 - * `hebcap` style should always be used (instead of older `hebcap_p` and `hebcap_newcode`).
 - * Font names for the included 7-bit fonts were renamed (for the sake of multiplatform support) to lowercase names with no more than 8 characters as following:

- o jerus10.mf - for Jerusalem (10pt)
- o deads10.mf - for Dead Sea (10pt)
- o telav10.mf - for TelAviv (10pt)
- o oldjaf10.mf - for Old Jaffa (10pt)
- o shold10.mf - for Shalom Old Style (10pt)
- o shscr10.mf - for Shalom Script (10pt)
- o shstk10.mf - for Shalom Stick (10pt)
- o redisb10.mf - for redisbx10.mf
- o Some other font-related files were renamed too (to lowercase and < 8 characters):
 - + frank_ruehl.mf became frruehl.mf
 - + redisfont.mf became redisfnt.mf
- o All other font names remain unchanged

Chapter 3

Notes on using vim

3.1 Introduction

GVIM, the graphical front end to VIM was chosen as the editor interface for several reasons. It's an advanced programmable editor with support for Hebrew, and what's most important, it is available with identical interface for most operating systems including Unix and Windows, thus using it gives us a uniform cross-platform interface. If Hebrew is not needed, another great option is to use Emacs with AUC-TeX.

Note that the version we use is VIM 5.3, some of the script will not work on earlier versions.

We provide some useful menus, and in addition many shortcuts that are useful for generating both Hebrew and non-Hebrew documents.

To use the functions provided by the \LaTeX menu, described in section 3.4, you will need the *TKlatex.vim*¹ and *menu.vim* distributed by us.

For the Win32 platform you will need (at least for version 5.3) the gvim executable compiled by us to handle Hebrew correctly².

If you compile VIM on Unix, enable (in features.h) the RIGHTLEFT, AUTOCMD and SYNTAX_HL options.

It is assumed the reader is familiar with the normal (English) use of VIM/GVIM. If not, you can start gvim and go over a short tutorial by entering

```
:e $VIM/tutor/tutor
```

After entering the “:”, the text you enter will appear on the bottom line. In case you changed /erased the tutor file, a backup copy of the tutorial is kept in the same directory as *tutor.sav*.

3.2 Fonts

¹ The features described here are available from version 0.32.

²This is due to the fact that reordering \LaTeX text with Hebrew according to the UNICODE guidelines results in a mess. The version we compiled mimics the Unix behavior – everything is printed left to right OR everything is printed right to left. See appendix ?? if you compile VIM yourself

3.2.1 Under X11

Naturally you have to use a font that contains the hebrew characters. The **heb8x13** is part of every X11 installation and can be used for this purpose.

VIM (non GUI) specify the font when opening the window. E.g.:

```
xterm -fn heb8x13
```

GVIM (GUI version) You have to set the *guifont* variable. I suggest to put the following line in your *.vimrc* file:

```
set guifont=heb8x13
```

See also appendix [A](#) for a sample *.vimrc* file.

3.2.2 Under Windows

If you have Hebrew enabled version of Win95 or NT, the system fonts used by default will do just fine. If you do not, you will have to install Hebrew truetype (monospaced) font and set the *guifont* variable accordingly. One example of a free TTF Hebrew font is *NeText* (provided by IBM Israel as the self-extracting file *NeText.exe*). To use it (after installation) put “set guifont=Netextmo” in your *.vimrc* file.

3.2.3 Problems with Hebrew fonts on Xwin-32

The “FON” fonts files aren’t working. Install the PCF version of the fonts (you can get the PCF from any Unix machine running X11).

3.3 Entering English and Hebrew

VIM displays the text “as is” and not in “logical” layout. I.e., the texts is either displayed left to right with Hebrew seems “reversed”, as in [Figure 3.1](#) or right to left with English looking reversed, as in [Figure 3.2](#).

This is one reason some users recommend switching lines every time you switch language. Most of the macros below ([section 3.5](#)) are set to do that.

There are editors, e.g., notepad, that display the text using logical layout, thus, theoretically, everything is readable. In practice, however, some of the things, especially braces (`{ }`), are not displayed correctly and can make finding errors very difficult.

3.3.1 Switching between English and Hebrew

Using the menu The *Language* menu toggles between Hebrew and English.

Using a function key When using the bindings in the sample *.vimrc* file ([Appendix A](#)) the `<F9>` key toggles between Hebrew and English.

Using the command line To set Hebrew mode use `:set hk` (set to hebrew keyboard) and `:set rl` (switch display direction). To return to English use `:set nohk` and `:set norl`.

```

VIM - ~/TeX/TKTEX/HebMan/VIMsample1.tex
File Edit Window Tools Syntax LaTeX Language Help
1 \begin{itemize}
2   \item You see an english text here
3   \item \R{טקסט ירביע רוזיא וזיא.}
4 \end{itemize}
5
6 \selectlanguage{hebrew}
7
8 ובו \emph{ירביע} רוזיא והז
9 \L{English text} סינכהל ותין
10
11 סיפידעמה שי
12 \emph{X}
13 דירפהל
14 }
15 תודרפינ תורושב תילגנאל תירביע ויב
~

```

Figure 3.1: VIM - English mode.

```

VIM - ~/TeX/TKTEX/HebMan/VIMsample1.tex
File Edit Window Tools Syntax LaTeX Language Help
1 \begin{itemize}
2   \item טקסט ירביע רוזיא וזיא.
3   \item \R{טקסט ירביע רוזיא וזיא.}
4 \end{itemize}
5
6 \selectlanguage{hebrew}
7
8 ובו \emph{ירביע} רוזיא והז
9 \L{English text} סינכהל ותין
10
11 סיפידעמה שי
12 \emph{X}
13 דירפהל
14 }
15 תודרפינ תורושב תילגנאל תירביע ויב
~

```

Figure 3.2: VIM - Hebrew mode.

3.3.2 Hebrew in the command line

In order to enter hebrew on the command line, e.g., to search for a word, you have to use *revin* (reverse input). This is toggled by `<CTRL-_->` (press the control, shift and the “-” keys while in the command line).

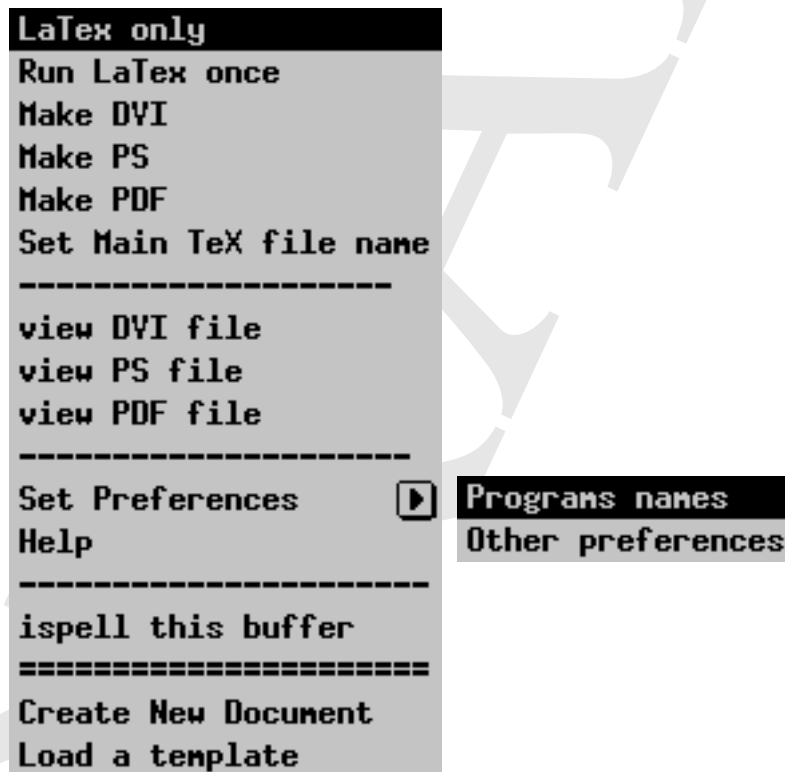
Note that you should toggle this back to English to enter commands such as `:w` etc.

If you use *revin* while editing, what you write will be pushed so that it will seem that english and hebrew looks in the right direction. **This is not what you want** (remember, beauty is only skin deep...), as \LaTeX reads all characters as a stream, so the case were either English OR Hebrew are displayed in the right direction is the correct setup.

Starting with VIM version 5.3, to enable this option (`<CTRL-_->`) you should have *set ari* (allow reverse insert) in your `.vimrc` file. E.g., see the sample `.vimrc` file in appendix A.

3.4 The \LaTeX menu

We configured VIM with a \LaTeX menu, shown in Figure 3.3, that gives the user access to most common tasks.



(a) The main \LaTeX menu

(b) The preferences sub-menu

Figure 3.3: The VIM \LaTeX menu

- **LaTeX only** – Runs only the \LaTeX program.
 - **Run LaTeX once** – Similar to the “Make DVI file” option, but run LaTeX makeindex etc. only once (links, cites etc. may be incorrect)
 - **Make DVI file** – runs LaTeX, makeindex and BibTeX enough times to solve references, citations etc.
 - **Make PS file** – create postscript file from the dvi file, tries to create the DVI file if it’s missing or old.
 - **Make PDF file** – create PDF file from the postscript file, tries to create the PS and DVI files if they are missing or old.
 - **Set Main Tex file name** – allows to set the name of the TeX file on which the following commands are executed. E.g. if you edit a file that is included into the main TeX file.
 - **View DVI file** – call the appropriate viewer.
 - **View PS file** – call the appropriate viewer.
 - **View PDF file** – call the appropriate viewer.
 - **Set preferences** –
 - **Programs names** Set the names of the programs called (E.g. view PDF using ghostview or Acrobat reader).
 - **Other preferences** Set various preferences
- NOTE:** Changes through this menu effects only the current session. For a permanent change set options in your .vimrc file, see docs.
- **Help** – Show a short help (this list).
 - **ispell this buffer** – run ispell on the current buffer. (ispell should be available on the computer)
 - **Create New Document** – Create a skeleton for a new \LaTeX document.
 - **Load a template** – open a file dialog allowing to load a template document (If you have created your favorite templates, store it in the templates dit and latter acces them through this menu).
 - **About** – Display version of *TKlatex* macros in use.

3.4.1 NOTES

- If the *Main Tex file name* is not set the name of the current buffer will be used. The main tex file name can be changed using the **Set Main Tex file name** menu option, e.g, when you edit a file included by the main tex file.
- If you open a tex file using ”Open new TeX document” from the File menu, *Main Tex file name* will be automatically set to the name of that file.

3.4.2 Errors while \LaTeX ing the file

whenever \LaTeX encounters an error, it will stop and ask you for input. Consult \LaTeX books for details, but as a start, entering ”X” (uppercase x) will usually cause \TeX to exit.

The *TKlatex* will offer you the dialog in figure 3.4.

- **Abort** – return to editing.
- **Load log file** – will load the log file into a split window.
- **Jump to Error** – will split the window and will (try to) load the relevent file, locating the cursor at the line indicated by \LaTeX as the casue of error.

Figure 3.4: VIM - Dialog presented after a \LaTeX error.

3.5 Sample macros and key bindings

The definitions are in effect if you use (i.e., place it in your home directory if using Unix³) the sample `.vimrc` file⁴ in appendix A.

For more information refer to the *VIM* documentation (available from within VIM by pressing `<F1>`).

Note that by default `<F1>` is bound to the online help.

`Ctrl-ENTER` - opens a new line with a language switch

`Shift-Enter` - Split current line at point, placing a line in the other language

`Shift-Control-ENTER` - make an insert of the other language
(`\L{}` or `\R{}`)

`Shift-END` - jump to end of line with a language switch, and enter append mode (usefull after an insert of the other language when `TKseparateLang` is not set.

`F2` saves the current buffer.

`F3` inserts a Latex enviroment (same as `<Ctrl-L><Ctrl-E>`)

`Shift-F3` inserts a Latex enviroment protected inside an english environment Needed for example for equation. (same as `<Ctrl-L><Shift-E>`)

`F4 / Shift-F4` create ref/label same as `<Ctrl-L><Ctrl-r>` and answering the questions.

`F5 / Shift-F5` create hyperlinks/targets same as `<Ctrl-L><Ctrl-r>` and answering the questions.

`F9` switches language without starting a new line.

³On windows NT/95 you can either place it in the VIM directory (it's there on the distribution we prepared) or define a `HOME` environment variable pointing to the directory you place it in

⁴On windows NT/95 the file is named `_vimrc`. If you have the environment variable "HOME" defined, vim will search for the file in the "HOME" directory. Otherwise it will look for it in the directory where vim is installed.

Shift-F9 switches language on the commandline.

3.5.1 Default L^AT_EX bindings

These bindings are installed by the line

```
call TKsetdefaultKeys()
```

in the *.vimrc* file.

Each is invoked by a sequence of two keys, pressed with a delay of less than 1 second in between⁵. <C-x> means that you have to press the Ctrl (control) key together with key x. Similarly <S-x> means that you have to press the Shift key together with the x key.

In most cases you will be queried for the parameters needed (type of sectioning command, placement of a figure etc.)

<C-l><C-d> Call *MakeDVI* to update *dvi* file (changed buffers are written to disk first).

<C-l><C-o> Run L^AT_EX (and bibtex etc.) *once*. Note that cross references may be unresolved. (changed buffers are written to disk first).

<C-l><C-s> Insert a sectioning command.

<C-l><C-e> Insert a L^AT_EX environment.

<C-l><S-e> Insert a L^AT_EX environment protected inside an English env (sometimes needed, for example, for an equation with a `\label`).

<C-l><C-r> Insert a command with text (e.g. `\ref{text}`). See also the bindings for F4 and F5 above and in appendix A.

<C-l><C-i> Insert a text in the other language inline (`\L{}` or `\R{}`)

<C-l><C-g> Insert a `\includegraphics` command.

<C-l><C-f> Insert a float (e.g., figure) environment.

<C-l><ENTER> Open a new line starting with `\item`.

3.5.2 Miscelenous useful options (not related to T_EX)

This is a list of some options and setting I find useful when using VIM. Naturally it's a matter of personal taste and habits. I've included it as it may be useful for novice users (if they share similar taste). This is no substitution for going over the documentation/tutorial.

Settings To negate add `no` to the option (i.e., put `:set nofoo` instead of `:set foo`).

- Case insensitive search `:set ignorecase`
- Incremental search `:set incsearch`
- Autoindent (start next line in same column as the current) `:set ai`
- Show line numbers `:set nu`

⁵This delay is configurable.

Useful keys

- u Undo (in command mode)
- ^R Redo (in command mode)
- . (period) repeat last operation (in command mode)
- / Search (in command mode)
- \% Jump to matching parenthesis (in command mode, curser on a parenthesis)
- ^X^I complete word (similar to M-/ in emacs)

DRAFT

Chapter 4

Notes on creating DVI/PS/PDF files

4.1 Hyperlinks – the *hyperref* package.

The *hyperref* package transforms normal L^AT_EX commands such as `\label`, `\ref` and `\cite` to create hyperlinks in the document. The hyperlinked document can be viewed by *xdvi*, or transformed into a PDF document and viewed using Adobe Acrobat Reader. See the following sections for further details.

For full description refer to its manual (see section 1.4).

NOTE: The *hyperref* version distributed is 6.56.

4.2 Converting an existing L^AT_EX document

If you have an (English) article written in L^AT_EX 2_ε that you want to make into an hyper-linked PDF document, all you have to add is the line

```
\usepackage[dvips,colorlinks,breaklinks]{hyperref} % Use for ps/pdf
```

You can also get an hyperlinked DVI document (hyperlinks are supported by recent versions of *xdvi*) by dropping the *dvips* option from the `\usepackage` line (note that in this case, if you will later convert this DVI file to PDF, *you will not get PDF hyperlinks*, to get it you'll have to recreate the DVI file with the *dvips* option).

Consider also using *pslatex* (see below) There are many other options such as adding keywords, controlling the appearance of links etc. Refer to Hyperref's documentation.

For **Hebrew** documents (using the *babel* package), you also have to include the *TKfixes* package, and make sure that the document conforms to all that is described above.

4.3 Putting additional document info (For Acrobat reader)

Put in the preamble:

```
\hypersetup{% Specific for Acrobat reader
  pdftitle=Using\ Hebrew\ LaTeX\ and\ gvim\ to\ produce\ DVI/PS/PDF,%
```



```

pdfauthor=Tomer\ Kol,%
pdfsubject=Hebrew\ Latex\,\ PDF\ creation.,%
pdfkeywords=LaTeX\ Hebrew\ PDF%
}

```

This information is displayed (in Adobe Acrobat Reader) through the menu File→Document Info→General, or by pressing CTRL-D.

4.4 The *showkeys* package

This package displays the keys of labels (as a side note) and references (above the reference). It is very handy during the creation of documents that contains references and citations.

To use it just put

```
\usepackage{showkeys}
```

After the `hyperref`'s `\usepackage` line.

4.5 The *pslatex* package

This package replaces the computer modern font with Adobe Acrobat's built in PS type 1 fonts (English and symbol fonts). It is highly recommended for any English documents as the resulting PDF will be much smaller and the English/math text of higher quality.

To use it just add to your document

```
\usepackage{pslatex}
```

Note: If you use it for an Hebrew document, you will lose the Hebrew bold and emphasized fonts. To use it anyway, place it **before** the `babel` line.

(The reason for this bug is known. This should be fixed in a future release)

4.6 Using Type 1 version of the Computer Modern fonts

To include the postscript version of the computer modern fonts in the postscript file¹ create (Unix) in your home directory a file named "`.dvipsrc`" that contains:

```
p +psfonts.amz
```

```
p +psfonts.cmz
```

For a systemwide change² you can add those two lines to the end of the `config.ps` file (found in `texmf/dvips/config`).

4.7 Hebrew in the bookmarks sidebar

In order to have Hebrew in the bookmarks sidebar, the `.out` file created by `hyperref` should be processed before running `LATEX` for the last time. This is done automatically when you call `MakeDVI` from VIM's `LATEX` menu.

¹This is highly recommended as it can make the postscript file smaller, and the PDF file looks **much** better.

²This is also the recommended approach for Win* users. it is already set up this way in our distribution.

Note Currently Hebrew (in the sidebar) doesn't show on Unix, and may also not show on versions of windows that are no Hebrew enabled. The main text shows on all platforms.

DRAFT

Chapter 5

Miscellaneous Problems and FAQ

The lines are not aligned This may show in normal text or in `itemize` or other list environments. Look in the `*.log` file for over-full lines. Over-full lines are spilled to the right (instead of to the left as one may expect). The solution, as with over full lines in English \LaTeX , is to change wording, put explicit newline etc.

Problem with the “`amssymb`” package The `amssymb` package defines the token `\gimel`, which is used by the Hebrew support¹. In our distribution, the AMS “`\gimel`” token is renamed to “`\AMSGimel`” (in the `amssyb.sty` file).

¹This is the name of third Hebrew letter. Other Hebrew letters defined by `amssymb` are spelled differently.

Appendix A

Sample .vimrc file

This is the **.vimrc** file in our distribution.

```
1  "+-----+
2  "|           .vimrc file
3  "|
4  "| Tomer Kol
5  "| Ver:  9 Mar  99
6  "+-----+
7
8  "
9  " Some useful settings (available in normal vi as well)
10 "
11 " Show line numbers
12 set number
13 "set autoindent
14 set sm
15 " Show current mode (insert, replace etc.)
16 set showmode
17 " case insensitive searches
18 set ignorecase
19 "
20
21 " Avoid warning for the new version - by popular demand.
22 version 5.3
23
24 " Automagically enable editing of gzipped files:
25 " read ->  set binary mode before reading the file, gunzip in
26 "         buffer after reading.
27 " write -> gzip after writing.
28 " append -> gunzip, append, gzip.
29 "
30 " From the sample vimrc in the upstream package.
31 " You may want to change gzip to gzip -9 for maximum compression -
32 " your mileage may vary. Experiment!
```

```

33
34 autocmd BufReadPre,FileReadPre          *.gz set bin
35 autocmd BufReadPost,FileReadPost        *.gz '[,']!gunzip
36 autocmd BufReadPost,FileReadPost        *.gz set nobin
37
38 autocmd BufWritePost,FileWritePost       *.gz !mv <file> <file>:r
39 autocmd BufWritePost,FileWritePost       *.gz !gzip <file>:r
40
41 autocmd FileAppendPre                    *.gz !gunzip <file>
42 autocmd FileAppendPre                    *.gz !mv <file>:r <file>
43 autocmd FileAppendPost                    *.gz !mv <file> <file>:r
44 autocmd FileAppendPost                    *.gz !gzip <file>:r
45
46
47
48 "
49 "
50 " set width for auto wrap
51 set textwidth=78
52
53 "
54 " comment the following line to disable syntax highlighting
55 syntax on
56
57 "+-----+
58 "|
59 "|           For HEBREW
60 "|
61 "+-----+
62 "
63 "
64 " select hebrew font for gvim
65 if ! has("win32")
66   set guifont=heb8x13
67 else
68   "set guifont=Netextmo
69 endif
70
71 " Allow revins (CTRL-_)
72 set ari
73
74
75 "+-----+
76 "|
77 "|   Keys bindings
78 "|
79 "+-----+

```

```

80
81 " Control-ENTER
82 "Open new line switching Language
83 map <C-CR> :set invrl<CR>:set invhk<CR>o
84 imap <C-CR> <Esc>:set invrl<CR>:set invhk<CR>o
85
86 " Shift-ENTER
87 "Split current line at point, placing a line in the other language
88 map <S-CR> i<CR><ESC>:set invrl<CR>:set invhk<CR>O
89 imap <S-CR> <Esc>i<CR><ESC>:set invrl<CR>:set invhk<CR>O
90
91 " Shift-Control-ENTER
92 " make an insert of the other language
93 imap <S-C-CR> <ESC>:call TKotherLangInsert()<CR>i
94
95 " Shift-END
96 " go to end of line, switching language
97 map <S-END> :set invrl<CR>:set invhk<CR>A
98 imap <S-END> <Esc>:set invrl<CR>:set invhk<CR>A
99
100
101 " FUNCTION KEYS
102
103 " <F1> is bound by default to help
104
105 " <F2> save buffer
106 map <F2> :w<CR>
107 imap <F2> <ESC>:w<CR>a
108
109 " For editing multiple files
110 map <F11> :prev<CR>
111 imap <F11> <ESC>:prev<CR>a
112 map <F12> :next<CR>
113 imap <F12> <ESC>:next<CR>a
114
115 " toggle both direction and hebrew keyboard mapping
116 map <F9> :set invrl<CR>:set invhk<CR>
117 " do it when in insert mode as well (and return to insert mode)
118 imap <F9> <Esc>:set invrl<CR>:set invhk<CR>a
119 "toggle comand line language
120 cmap <S-F9> <C-_>
121 " toggale language and add at EOL (useful after CLCI use CLCB instead?)
122 map <C-F9> :set invrl<CR>:set invhk<CR>
123 " do it when in insert mode as well (and return to insert mode)
124 imap <C-F9> <Esc>:set invrl<CR>:set invhk<CR>A
125
126 "insert LaTeX envioment

```



```
174 " let TKtemplatesDir = "~/TeX/Templates/"  
175 endif
```

DRAFT

Appendix B

Sample L^AT_EX files

These are simple L^AT_EX files. Examples supplied by Boris Lavva can be found in the distribution under /texmf/doc/Hebrew.

B.1 Sample English only document.

```
1 %+=====+
2 %|      Skeleton created by TKlatrix      |
3 %+=====+
4 \documentclass[a4paper,12pt]{article}
5 %+-----+
6 %|      Packages      |
7 %+-----+
8 % Next line is needed if you intend to include graphic files
9 % in your document.
10 \usepackage{graphicx}
11 % The next two lines, and the "\printindex" line in the end,
12 % are needed if you want your document to have an index.
13 \usepackage{makeidx}
14 \makeindex
15
16 % The hyperref line is needed if you want to use hyperlinks.
17 % Hyperref is configured for PS and PDF files.
18 % To get hyperlinked DVI remove the "dvips" option
19 % of hyperref.
20 \usepackage[dvips,colorlinks,breaklinks]{hyperref}
21
22 %+=====+
23 %|      Doc content !!      |
24 %+=====+
25 \begin{document}
26
27 % The next line creates the index.
```

```

28 \tableofcontents
29
30 %The folowing lines are an example of using \label and \ref
31 % (For beginners: note that they are commented out)
32 %=====
33 %\section{First Section}
34 %\label{sec:intro}
35 %
36 %This is section~\ref{sec:intro}.
37
38
39 %next command includes the index
40 \printindex
41 \end{document}
42

```

B.2 Sample Hebrew/English document.

```

1  %+=====+
2  % |      Skeleton created by TKlatex      |
3  %+=====+
4  \documentclass[a4paper,12pt]{article}
5  %+-----+
6  % |      Packages      |
7  %+-----+
8  % Next line is needed if you intend to include graphic files
9  % in your document.
10 \usepackage{graphicx}
11 % The next two lines, and the "\printindex" line in the end,
12 % are needed if you want your document to have an index.
13 \usepackage{makeidx}
14 \makeindex
15
16 % The next line, and the "TKlatex" line below, calls the Hebrew
17 % support. The last language specified (Hebrew in this case) will
18 % be the default language in the document.
19 \usepackage[english,hebrew]{babel}
20
21 % The hyperref line is needed if you want to use hyperlinks.
22 % Hyperref is configured for PS and PDF files.
23 % To get hyperlinked DVI remove the "dvips" option
24 % of hyperref.
25 \usepackage[dvips,colorlinks,breaklinks]{hyperref}
26
27 % TKfixes should be the last package

```

```
28 \usepackage{TKfixes}
29
30 %+=====+
31 %|          Doc content !!          |
32 %+=====+
33 \begin{document}
34
35 % The next line creates the index.
36 \tableofcontents
37
38
39
40 %next command includes the index
41 \printindex
42 \end{document}
43
```

Index

dvipsrc, 24

graphics/graphicx (package), 10

GVIM, *see* VIM

Hebrew Latex, using the package, 6

hyper-links, *see* hyperref

hyperref (package), 23

Including figures, *see* graphics/graphicx
index, 9

Known problems, 26

LaTeX

resources, 4

templates, 32

PDF

adding additional document info, 23

converting existing documents, 23

using type 1 fonts, *see* pslatex and dvipsrc

pslatex (package), 24

showkeys (package), 24

theorem like environment in Hebrew, 9

TKfixes, 6, 23

VIM, 15, 15–22

and fonts, 15

customization, 20

English and Hebrew, 16

initialization, *see* VIM, vimrc

LaTeX menu, 18

vimrc, sample, 27