Software Manual
for
Windows Z/EVES Version 2.3

Contents

1   Introduction

This document describes the procedure for installing Version 2.3 of the Z/EVES verification system on a Windows 98, Windows NT, Windows 2000, or Windows XP system. The distribution format for Windows Z/EVES is a self-installing executable file, usually downloaded using FTP.

This version of Z/EVES can be used in two ways, in server mode and in Z/LaTeX mode. In server mode, it runs as a server, providing parsing, type checking, and proof service which is accessed through a graphical user interface (GUI). The GUI is included in the Z/EVES distribution and is described in [Saa 99b]. The Z/EVES GUI accepts and displays Z text in Z notation, as described in the Z Reference Manual ([Spi 92]).

In Z/LaTeX mode, Z/EVES is accessed directly, through a command-line interface provided by Z/EVES itself. In this mode, Z text is accepted and displayed in Z/LaTeX notation. Also, a GNU Emacs mode is provided for editing Z/LaTeX text and interacting with Z/EVES.

The Z/EVES distribution includes the Z font created by Richard Jones, Version 1.0.4; this font is ©Richard Jones, 1995.

1.1   GUI Requirements

The Z/EVES GUI is written in Python, and requires Python 2.3 for Windows; an executable file which installs this program is available from `http://www.python.org/2.3.4/`.

We have experienced some problems with Python 2.3.x on Windows 98 machines with limited memory (32Mb on our test machine) and do not recommend running Z/EVES 2.3 on such machines.

2   Release Notes

Version 2.2 had a limited, Windows-only distribution; the last major public release of Z/EVES was Version 2.1. Save files created with Version 2.2 can be read by Version 2.3. However, because the Toolkit has changed, save files created by version 2.1 and earlier versions of Z/EVES may be read, but the proofs are lost and proof scripts, if any, must be run again. The following changes have been made since Version 2.1:

- Windows Z/EVES is now built with Xanalys LispWorks, which is considerably faster than the Lisp system used previously.

- The Z/EVES GUI will now work with a more up-to-date version of Python (Version 2.3).

- Save and restore will now work for very large files.

- The mouse wheel, if you have one, now works correctly in the specification and proof windows.

- A specification can be exported in LaTeX form. This facility has some minor shortcomings:

  - Unchecked syntax declarations are exported incorrectly.
  - Some symbols are exported incorrectly:

symbol	exported as
\upto	..
\hide	\setminus
\comp	\semi

    In some cases, a `\Label{...}` command can be exported as `\ldata` ... `\rdata` instead. In all these cases, the exported text can be printed or re-imported without problems, but cannot be successfully checked in the text mode Z/EVES interface.
  - Generic actuals for schemas appear in the wrong place.

  – A paragraph's "ability" can sometimes be lost.

- The Z/LaTeX form allows for proof scripts to appear in a "zproof" environment. This has the syntax:

  > Zproof   ::=   \begin{zproof} [goal-name] command ; ... ; command \end{zproof}
  > goal-name   ::=   [event-name]

  The `read` command in the text-mode interface to Z/EVES has been modified to process these environments. If no goal-name is given in the environment, the commands are applied to the current goal. Thus, a zproof environment appearing immediately after a theorem will apply to that theorem.

  In the text-mode interface to Z/EVES, several new commands are applicable to files containing zproof environments:

  – The `read-declarations` command reads and processes any Z paragraphs in the given file, skipping the contents of any zproof environments.
  – `read-proofs` reads and processes and zproof environments in the given file.

  The GUI's LaTeX import function reads zproof environments, so that proofs developed in text mode can be imported into, and re-run in, the GUI. The LaTeX export function can also generate these environments.

- A new command, "Commands|Check and prove all paragraphs", has been added. This command checks paragraphs and runs proof scripts for goals as they are generated, and is particularly useful for re-running proofs after making changes to a specification.

- A new command, "File|Properties", giving some summary information about the specification, has been added.

- A new error, "Ill-formed binding" can be reported if a proof step results in an illegal Z formula and the proof step has no effect. Previously, the system would crash.

- A few glyphs have been moved around in the Z font, so that they will be displayed correctly in new versions of Python. The font has been renamed so as not to conflict with the unmodified version.

- The Toolkit has been modified:

  – Many new theorems have been added. There are significant additions for constant functions, transitive closure ($\_^+$), reflexive transitive closure ($\_^*$), and arithmetic.
  – Some typos in theorem names have been fixed:

old name	new name
$CrossSubsetCross2$	$crossSubsetCross2$
$CrossSubsetCross3$	$crossSubsetCross3$
$cardIsNonNegqative$	$cardIsNonNegative$

  – Theorems $crossSubsetCross2$, $crossSubsetcross3$, and $mapSeqCat$ have been strengthened.
  – Theorem $starDef$ has been corrected.
  – The defining axioms for prefix, suffix, and in have been labelled, so they can be more easily used in proofs. This as a side-effect means a user's first unnamed axiom will be $axiom\$1$ instead of $axiom\$4$, so some existing proofs might be affected.

- New internal rules have been added to allow rewrite steps to deal with declarations of the form $x : \{\}$ or $x : \{y\}$.

- Several errors have been fixed:

    – A bug has been fixed in the LaTeX import function; this bug was triggered when a `.tex` file was imported, a paragraph was deleted in the GUI, and then the `.tex` file was imported again.

    – The LaTeX importer now handles `div` and `mod` correctly.

    – Applications of prefix generics are now properly parenthesized when necessary.

    – Bindings and var-decls are now printed correctly.

    – The alphabets of a binding are now sorted, so that the order in which components are mentioned does not matter.

    – Two problems with variable renamings have been corrected.

3   Installation Instructions

Before installing Z/EVES, you must install Python 2.3.4 if you wish to use the Z/EVES GUI.

3.1   Installing Python

To install Python, do the following:

1. Download the Python 2.3.4 installer (`Python-2.3.4.exe`) at the URL given in Section 1.

2. Run the installer. On Windows NT or above, log in as Administrator first.

3.2   Installing Z/EVES

The Windows Z/EVES distribution file is an executable file which installs Z/EVES and the Jones Z font, creates a program group for Z/EVES 2.3, and creates shortcuts on the desktop for running Z/EVES. The installer also creates links, in the Z/EVES program group, to the Z/EVES documentation files. These files are in Adobe PDF format, and you need a PDF reader to view them.

   To install Z/EVES, do the following:

1. Download the Z/EVES installer via FTP, to some temporary directory. Make sure that you download the correct version for your system, and that you download the file in binary mode. (For more specific instructions on how to download Z/EVES, please contact us at the address given in Section 6.)

2. Double-click on the installer file in Windows Explorer, or run it from the Start menu. Administrator privileges are not required.

3. The installer will create a program group for Z-EVES 2.3 accessible from the Start menu, and install shortcuts for the Z/EVES command line and Z/EVES GUI on the desktop. By default, Z/EVES files are installed in subdirectories of `C:\Program Files\Z-EVES 2.3\`.

4. [optional] The file `system\z-eves.el` in the installation directory contains Emacs Lisp code for the GNU Emacs Z/LaTeX and Z/EVES modes. This code works only with GNU Emacs Version 19 and above. To install and compile this file, do the following:

   (a) Copy `z-eves.el` to your Emacs Lisp site directory.

   (b) Visit the file in Emacs, and use `M-x byte-compile-file` to compile the file. You will get some warnings about assignments and references to free variables; these may be ignored.

   (c) Add the following to your `~/.emacs` file:

```
(setq auto-mode-alist (append auto-mode-alist
                              '(("\\.z$" . z-latex-mode)
                                ("\\.zed$" . z-latex-mode))))
(autoload 'z-latex-mode "z-eves" "Z-EVES LaTeX mode." t)
(autoload 'run-z-eves "z-eves" "Run Z-EVES." t)
(setq z-eves-program
 "c:/program files/z-eves 2.3/system/z-eves-pc-windows-lispworks.exe")
(setq z-eves-program-args nil)
```

   This assumes that Z/EVES has been installed in the default location. If not, specify the full path name of the Z/EVES executable as the value of `z-eves-program`.

   The modifications to `auto-mode-alist` will make Emacs automatically turn on Z/LaTeX mode in a buffer for a file whose extension is `.z` or `.zed`. For a file whose extension

is `.tex`, you will have to turn on Z/LATEX mode in the buffer manually. You can add the cons (`"\\.tex$" . z-latex-mode`) to `auto-mode-alist` to automatically turn on Z/LATEX mode for these files, but this will interfere with the editing of TEX or LATEX source files.

3.3 Uninstalling Z/EVES

Z/EVES may be uninstalled from the Z-EVES 2.3 program group, or from the Add/Remove Programs applet in the Control Panel.

3.4 The Toolkit and Section Directories

NOTE: Sections are fully implemented only in Z/LATEX mode. The Z/EVES server starts up with the Toolkit section loaded, and it is not possible to replace this section or load other sections.

The `system\` subdirectory of the Z/EVES installation directory contains the section file for the Z/EVES Mathematical Toolkit, `toolkit.sct`. Z/EVES starts with the Toolkit section already loaded, but if Z/EVES needs to load the Toolkit again, it must know where to find the section file.

Z/EVES looks for section files in the following directories, in the following order:

1. The directory which contains the Z/EVES executable (the "system" directory).

2. Directories specified in the Z/EVES command line, with the `-libs` option. The value of this option is a semicolon-separated list of directories. This option may be added to the command line in the "Z-EVES" shortcut. (As noted in Section 4.1, this option must be preceded with `--` if it is the first Z/EVES option.)

3. Directories specified in the `zsection path` command, in the order specified (the "user" directories).

In the second and third cases, directory names must end in a backslash (`\`).

If the `zsection path` command is given with no arguments, the current system directory and user directory list are printed.

4 Using Windows Z/EVES

To run Z/EVES in Z/LATEX mode, double-click on the "Z-EVES" icon on the desktop, and a command-line window will appear. To run the Z/EVES GUI, double-click on the "Z-EVES GUI" icon on the desktop, and the Z/EVES GUI main window will appear. These shortcuts are also available in the Z-EVES 2.3 program group.

Chapter 2 of the User's Guide [Saa 99b] describes how to use the GUI.

To use Z/EVES effectively, you will need to have a good knowledge of the Z language and the syntax of Z/EVES input. The Z reference manual [Spi 92] describes the former, and the Z/EVES reference manual [Mei 97] describes the latter. When proving theorems and domain checking conditions, you will find the theorems described in the documentation for the Z/EVES Mathematical Toolkit [Saa 99a] useful.

4.1 Z/EVES Program Options

The Z/EVES program accepts several options. The first Z/EVES option must be preceded by a double hyphen (`--`). The following options are accepted:

-help Print a short description of the valid options and exit.

-version Print version information and exit.

-api Run Z/EVES in server mode (used by the GUI).

-libs dirs Specify a semicolon-separated list of directories in which to look for section files.

4.2   Using Unix Z/EVES with GNU Emacs

There are GNU Emacs major modes for editing Z/EVES source and for running Z/EVES. These modes are available if you have installed the `z-eves.el` Emacs Lisp file and modified your `~/.emacs` file appropriately (Section 3.2). When editing a file whose name ends in `.z` or `.zed`, the buffer will automatically be placed in Z/L^AT_EX mode.

The following commands are available to start Z/L^AT_EX mode in a buffer, and to start an inferior Z/EVES process.

`M-x z-latex-mode`	start Z/L^AT_EX mode
`M-x run-z-eves`	start an inferior Z/EVES process

The following commands are available in Z/L^AT_EX mode:

`C-c C-l`	start an inferior Z/EVES process or pop to existing process
`C-c C-b`	move to the previous command or declaration
`C-c C-f`	move to the next command or declaration
`DEL`	delete previous char; if a tab, convert to spaces first
`C-j`	insert a newline and indent to previous line
`TAB`	insert a tab, or complete symbol before point

For a Z paragraph to be recognized by Z/L^AT_EX mode, the `\begin` and `\end` text must start in the first column of a line. The text of a command must also start in the first column of a line.

When an inferior Z/EVES process is started, a window connected to the process is popped up, and the following additional commands are made available in Z/L^AT_EX mode:

`C-c C-e`	send the declaration or command which starts at or before point to Z/EVES
`C-c C-r`	send declarations and commands in the current region to Z/EVES
`C-c C-s`	insert a proof summary at point

The following command is available in the Z/EVES process window:

`C-c C-c`	interrupt Z/EVES

If the `C-c C-r` command is given a numeric argument (using the `C-u` prefix), only declarations and non-proof commands in the region will be sent to the Z/EVES process. Also, if the `C-c C-s` command is given a numeric argument, the inserted proof summary will be enclosed in a zproof environment.

Output from the form(s) sent to the Z/EVES process by `C-c C-e` and `C-c C-r` will appear in the Z/EVES window. If the `C-c C-l` or `M-x run-z-eves` command is given while a Z/EVES process is running, the Z/EVES window will be popped up on the screen if it is not currently visible. You can also move to the Z/EVES window and type Z/EVES commands there; it is a "comint window" and the commands available in comint mode are also available in the Z/EVES window.

If you have used the `ztags` command to generate a tags file for your Z/L^AT_EX source file(s), the tags-related Emacs commands, (e.g., `M-x tags-search`) may be used, but note the following:

- The current buffer must be in Z/L^AT_EX mode when the tags table is loaded. If not, you will have to kill the tags table buffer and reload the table.

- To search for a name or symbol defined in a `\syndef` declaration, enclose the name or symbol in double quotes (`"`), to distinguish it from the name defined in a declaration box.

The Z/EVES process may be terminated by exiting Z/EVES and killing the Z/EVES window.

5 LaTeX 2ε Support

The Windows Z/EVES distribution includes a LaTeX2e package, `z-eves`, for typesetting Z/LaTeX specifications. This package consists of the Z macros from the `zed-csp` package written by Jim Davies, plus macros for typesetting the Z/LaTeX extensions to Z (e.g., theorems).

To install this package, copy the file `doc\z-eves.sty` to your site-local LaTeX package directory.

6   Contact

If you have any problems with the system, contact us at:

Postal mail:

```
Z/EVES
c/o ORA Canada
P.O. Box 46005
2339 Ogilvie Road
Ottawa, Ontario K1J 9M7
```

Electronic mail:

```
eves@ora.on.ca
```

World Wide Web:

```
http://www.ora.on.ca/
```

There is a mailing list (email) for Z/EVES-related announcements and news, and for general discussion about Z/EVES and Z. The list is a good way to keep in touch with the Z/EVES developers and with Z/EVES users.

- To subscribe to the list, send an email message to `zeves-request@ora.on.ca`, with no subject and the word `subscribe` in the message body.

- To unsubscribe from the list, send an email message to `zeves-request@ora.on.ca`, with no subject and the word `unsubscribe` in the message body.

- To post a message to the list, email the message to `zeves@ora.on.ca`. *Please* don't send subscription/unsubscription requests to this address.

- The Z/EVES mailing list is implemented with Majordomo. For general information on Majordomo functionality, send an email message to `majordomo@ora.on.ca`, with no subject and the word `help` in the message body.

- For help with problems, send an email message to `postmaster@ora.on.ca`.

References

[Mei 97]   Irwin Meisels and Mark Saaltink. *The Z/EVES Reference Manual*. ORA Canada Technical Report TR-97-5493-03d, September 1997.

[Saa 99a]   Mark Saaltink. *The Z/EVES 2.2 Mathematical Toolkit*. ORA Canada Technical Report TR-03-5493-05c, June 2003.

[Saa 99b]   Mark Saaltink. *The Z/EVES 2.0 User's Guide*. ORA Canada Technical Report TR-99-5493-06a, October 1999.

[Spi 92]   J.M. Spivey. *The Z Notation: A Reference Manual*, 2nd Ed. Prentice Hall, 1992