

**Software Manual
for
Unix Z/EVES Version 2.3**

TR-00-6028-01f

Irwin Meisels

Release date: June 2004

ORA Canada
P.O. Box 46005, 2339 Ogilvie Rd.
Ottawa, Ontario K1J 9M7
CANADA

Contents

1	Introduction	1
1.1	GUI Requirements	1
2	Release Notes	1
3	Installation Procedure	4
3.1	The Toolkit and Section Directories	5
4	Using Unix Z/EVES	7
4.1	Z/EVES Program Options	7
4.2	Using Unix Z/EVES with GNU Emacs	7
5	L^AT_EX 2_ε Support	9
6	Contact	10

1 Introduction

This document describes the procedure for installing Version 2.3 of the Z/EVES verification system on a Linux system. The distribution format for Unix Z/EVES is a compressed tar archive, usually downloaded using FTP.

This version of Z/EVES can be used in two ways, in server mode and in Z/L^AT_EX 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/L^AT_EX 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/L^AT_EX notation. Also, a GNU Emacs mode is provided for editing Z/L^AT_EX text and interacting with Z/EVES.

This version of Z/EVES should run on Red Hat Linux versions 7.2 to 9, and on other distributions. Z/EVES does not yet work on Fedora, since the Lisp it is built with (CMUCL) does not.

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; versions of the GUI are provided for Python versions 1.5.2, 2.2.x, and 2.3.x. The version of Python must also have been compiled with support for threading and for using Tcl/Tk (Tkinter), version 8.0.4 or later. The version of Python that comes with Red Hat Linux 7.2 or later is usually sufficient.

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:

- The GUI can be run in more versions of Python. Versions 1.5.2, 2.2.x, and 2.3.x should work.
- Z/EVES is now built with CMUCL, which is considerably faster and more memory-efficient than GCL.
- The mouse wheel, if you have one, now works correctly in the specification and proof windows.
- A specification can be exported in L^AT_EX form. This facility has some minor shortcomings:
 - Unchecked syntax declarations are exported incorrectly.
 - Some symbols are exported incorrectly:

symbol	exported as
<code>\upto</code>	<code>..</code>
<code>\hide</code>	<code>\setminusminus</code>
<code>\comp</code>	<code>\semi</code>

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/L^AT_EX 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 L^AT_EX import function reads zproof environments, so that proofs developed in text mode can be imported into, and re-run in, the GUI. The L^AT_EX 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
<i>CrossSubsetCross2</i>	<i>crossSubsetCross2</i>
<i>CrossSubsetCross3</i>	<i>crossSubsetCross3</i>
<i>cardIsNonNegqative</i>	<i>cardIsNonNegative</i>
 - 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 Procedure

1. Download the Z/EVES distribution file via FTP. 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. Create and go to an installation directory. As long as there is enough space (about 50 MB), it doesn't matter where the directory is located. In the following, `<zeves>` is the full pathname of this directory.
3. Extract the Z/EVES system files from the distribution file. The tar file has been compressed with GNU gzip; GNU tar will automatically uncompress it if you specify the `z` option:

```
tar xvfz file.tgz
```

If you don't have GNU tar, uncompress it to standard output and pipe the result to tar:

```
gzip -cd file.tgz | tar xvf -
```

4. Make a copy of the file `system/z-eves.sh` named `z-eves`. Edit the file `z-eves` to set the `zevesdir` variable to the Z/EVES directory `<zeves>`. Move the `z-eves` file to a directory on your search path, and make sure that it is executable.
5. Install the GUI shell script and Z font:

- (a) Run the Python interpreter, note its version (in the startup banner), and hit Control-D to exit the interpreter. Make a copy of the file `system/z-eves-gui.sh` named `z-eves-gui`. Edit `z-eves-gui` to set the `zevesdir` variable to the Z/EVES directory `<zeves>`, to set the `zguidir` variable to the GUI directory that corresponds to your version of Python (`gui-1.5/`, `gui-2.2/`, or `gui-2.3/`), and to set the `python` variable to the location of the Python interpreter, if required. Move the `z-eves-gui` file to a directory on your search path, and make sure that it is executable.
- (b) The directory `zedfont/` contains the Jones Z font in PostScript Type1 and TrueType formats. (The TrueType font should work for X servers and X font servers that have FreeType support, but we have not tried this.) This directory also contains the `fonts.dir` and `fonts.scale` files required by the X server or X font server. In the following instructions, `<zefont>` is the full path to the font directory (`zedfont/`).

The Z font may be installed as a user font or as a system font. To install the Z font as a user font, run the command

```
xset fp+ <zefont>
```

every time the X server is started. This may be done manually, or the command may be added to your X initialization file (typically, `~/.xinitrc`).

Installation as a user font is recommended, because no root privilege is required, and you don't need to know system-dependent details about your X server (e.g., whether or not the X font server is used, and where the configuration files are). However, if you want to install the Z font as a system font, and your system is running the X font server, copy (as root) the `<zefont>` directory to somewhere in `/usr` (e.g., `/usr/local/fonts/zedfont`) and tell the font server about it:

```
/usr/sbin/chkfontpath -a /usr/local/fonts/zedfont
```


Note that the `chkfontpath` program appears to ignore add commands for directories not in `/usr`.

Alternatively, you can configure the X server directly to use the font. Edit (as root) the X server configuration file (usually something like `/etc/X11/XF86Config-4`) and add a `FontPath` directive for the font directory. Also, ensure that the module `type1` is loaded. Then, restart the X server.

6. [optional] The file `system/z-eves.el` contains Emacs Lisp code for the GNU Emacs Z/L^AT_EX and Z/EVES modes. To use these modes, we recommend that you byte-compile the file first. The Z/L^AT_EX and Z/EVES modes work only with GNU Emacs Version 19 and above. To compile and install this file, do the following:

- (a) Go to the `system/` directory and run Emacs on the file `z-eves.el`:

```
cd system
emacs -batch --eval '(byte-compile-file "z-eves.el")'
```

You will get some warnings about assignments and references to free variables; these may be ignored. You can also edit `z-eves.el` with Emacs and byte-compile it from within the editor with `M-x byte-compile-file`; do this if the nested quotes in the above command confuse your shell.

- (b) Copy the `z-eves.el` and resulting `z-eves.elc` files to your Emacs site-lisp directory.
- (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 "z-eves")
(setq z-eves-program-args nil)
```

This assumes the shell script `z-eves` has been moved to a directory on your search path. If not, specify the full path name of the shell script file as the value of `z-eves-program`. The modifications to `auto-mode-alist` will make Emacs automatically turn on Z/L^AT_EX 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/L^AT_EX mode in the buffer manually. You can add the cons `("\\.tex$" . z-latex-mode)` to `auto-mode-alist` to automatically turn on Z/L^AT_EX mode for these files, but this will interfere with the editing of T_EX or L^AT_EX source files.

3.1 The Toolkit and Section Directories

NOTE: Sections are fully implemented only in Z/L^AT_EX 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 directory `library/` contains the section file for the Z/EVES Mathematical Toolkit. Z/EVES starts with the Toolkit 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. Directories in the “system” library directory list, which is passed to Z/EVES by the `z-eves.sh` shell script. This is a colon-separated list of directories; Z/EVES searches each directory in the list, in the order specified. In the distributed `z-eves` script, this list contains only the directory `$zevesdir/library/`, and can be changed by changing the definition of the `zlibdir` variable in the script.

2. Directories in the “user” library directory list, which is specified in the `zsection path` command. Z/EVES searches each directory in the list, in the order specified.

Directory names must be specified with a trailing slash, in both the `z-eves` shell script, and in the `zsection path` command.

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

4 Using Unix Z/EVES

To run Z/EVES in Z/L^AT_EX mode, run the Z/EVES shell script `z-eves`. Z/EVES will type an introductory message, and present you with its top-level prompt `>`. Z/EVES is now in “interactive” mode, and accepts declarations and commands. To abort a proof command that is taking too long or to stop Z/EVES when printing a too-long formula, use the Unix interrupt key `^C` (control-C). Z/EVES will stop whatever it is doing and return to the top level.

To run the Z/EVES GUI, run the Z/EVES shell script `z-eves-gui`. 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 colon-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). 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/LATEX 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 2 ϵ Support

The Unix Z/EVES distribution includes a \LaTeX 2 ϵ 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. You may also have to run the **texhash** program so \LaTeX will be able to find this file.

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