NAME

dfrotz − interpreter for Infocom and other Z-Machine games (dumb interface)

SYNOPSIS

dfrotz [*options*] *file* [*blorb_file*]

At least one file must be specified on the command line. This can be either a plain Z-code file or a Blorb file. A Z-code file is a compiled executable for the Z-Machine. A Blorb file contains audio, graphics, and other things in addition to the game wrapped up into a single file. It can also optionally contain the Z-Machine executable. If a plain Z-code file is supplied, then**Fr otz** will check for a Blorb file with the same base name but an extension of **.blb, .blorb,** or **.zblorb** and load it if found.

If the file supplied on the command line is a Blorb file, then **Frotz** will check to see if a Z-code file is contained within. If not found, then **Frotz** will complain and exit.

An alternatively-named Blorb file can be supplied as the optional second parameter to the command line invocation.

DESCRIPTION

Frotz is a Z-Machine interpreter. The Z-machine is a virtual machine designed by Infocom to run all of their text adventures. It went through multiple revisions during the lifetime of the company, and two further revisions (V7 and V8) were created by Graham Nelson after the company's demise. The specification is now quite well documented; this version of Frotz supports version 1.0.

This version of Frotz fully supports all these versions of the Z-Machine except for version 6. Version 6 is semi-supported by displaying the outlines of V6 graphics with the picture number in the bottom-right corner.

This manpage is for Frotz with the dumb interface. Dumb Frotz does not support sound-effects, graphics, or colors.

OPTIONS

−a Watch attribute setting. Setting and clearing of attributes on objects will be noted in debugging messages.

−A Watch attribute testing. Every time the Z-machine tests an attribute value, the test and the result will be reported.

−f [irc | ansi | normal]

Select type of format codes. Dumb Frotz can optionally mark up its output with codes for color, bold, underline, and so on. Currently, formatting codes for **irc** clients and **ansi** terminals are supported. Specifying**normal** or not using this flag at all will result in Dumb Frotz' s normal behavior of not using any sort of markup.

−h N Text height. Every N lines, a MORE prompt will be printed. Use of the **−m** option renders this option moot.

−i Ignore fatal errors. If a Z-Machine interpreter encounters a zcode error such as division-by-zero or addressing an illegal object, the proper response is to abort execution. This is done because the zcode program doesn't have a clear idea of what is going on. There are some games out there that cause fatal errors because the authors were careless and used an interpreter that didn't properly check for errors. This option is intended to get around such bugs, but be warned that Strange Things may happen if fatal errors are not caught.

−I N Set the interpreter number. Infocom designed the Z-machine such that a game could tell on what kind of machine the interpreter was running. See INTERPRETER NUMBER below.

−L <filename>
When the game starts, load this saved game file.

−m Turn off MORE prompts. This can be desirable when using a printing terminal.

−o Watch object movement. This option enables debugging messages from the interpreter which describe the moving of objects in the object tree.

−O Watch object location. These debugging messages detail the locations of objects in the object tree.

−p Plain ASCII output only. This inhibits the output of accented letters and other characters from the Latin-1 character set, replacing them with reasonable alternatives. This may be necessary on devices lacking these characters. The OE/oe dipthongs are missing from the Latin-1 set. These are handled as well.

−P Alter the piracy opcode. The piracy opcode was never used by Infocom. This switch is really only useful for those who like to toy around with Z-code.

−r xxx Set runtime options. This option may be used repeatedly. See**CONFIGURA TION** below. There is no need to prepend a backslash.

−R <path>
Restricted read/write. Reading and writing files will be restricted only to the provided path. Ordinarily Frotz will write or read its saves, transcripts, and move recordings in whatever path or directory the user provides when the **SAVE, SCRIPT,** or **RECORDING** commands are given. This can be undesirable if Frotz is run in a restricted environment, by a front end, or by a chatbot. This option will cause Frotz to write or read only to the provided path and nowhere else. Then the controlling process can then watch that directory for changes and need not worry about someone scribbling or snooping who-knows-where.

−s N Set the random number seed value. The given seed value is used as the initial seed value on every restart. This is helpful for testing games like **Curses** which make random decisions before the first input (such that the hot key Alt−S does not really help).

−S N Set the transcript width. By default your transcript files are formatted to a width of 80 columns per line, regardless of the current text width. This switch allows you to change this setting. In particular, use −S 0 to deactivate automatic line splitting in transcript files.

−t Sets the Z-machine's *Tandy bit,* which may affect the behavior of certain Infocom games. For example, Zork I pretends not to have sequels, and Witness has its language toned down.

−u N Sets the number of slots available for Frotz's multiple undo hotkey (see below). This defaults to twenty, which should be sufficient for most purposes. Setting too high a number here may be dangerous on machines with limited memory.

−w N Manually sets the text width. This should not be necessary except in special circumstances.

−x Expand the abbreviations "g", "x", and "z" to "again", "examine", and "wait". This switch is for use with old Infocom games that lack these common abbreviations which were introduced in later games. Use it with caution: A few games might use "g", "x" or "z" for different purposes.

−v Show version information and exit. This will display the version of Frotz, some information about what's enabled and what's not, the commit date of the source code, and a git(1) hash of that commit.

−Z N Error checking mode.
0 = don't report errors.
1 = report first instance of an error.
2 = report all errors.
3 = exit after any error.
Default is 1 (report first instance of an error).

INTERPRETER NUMBER

The interpreter number is a setting in the Z-machine header which is used to tell the game on what sort of machine the interpreter is running. Frotz will automatically choose the most appropriate number for a given Infocom-produced game. Should you want to override the number, the **−I** option is available.

An interpreter should choose the interpreter number most suitable for the machine it will run on. In Versions up to 5, the main consideration is that the behaviour of 'Beyond Zork' depends on the interpreter number (in terms of its usage of the character graphics font). In Version 6, the decision is more serious, as existing Infocom story files depend on interpreter number in many ways: moreover, some story files expect to be run only on the interpreters for a particular machine. There are, for instance, specifically Amiga versions. The DECSystem-20 was Infocom's own in-house mainframe.

For Infocom's four V6 games, the interpreter number will be automatically chosen based on the title and release number. Of course, this can be overridden at the command line.

Infocom used the following interpreter numbers:

1 DECSystem 20

2 Apple IIe

3 Macintosh

4 Amiga

5 Atari ST

6 IBM PC

7 Commodore128

8 Commodore64

9 Apple IIc

10 Apple IIgs

11 Tandy Color

CONFIGURATION

Unlike it's curses-using or SDL-using siblings, **dfrotz** does not use configuration files. All configuration is done on the command line or while **dfrotz** is running.

General Commands

\help Show help message.

\set Show the current values of runtime settings.

\s Show the current contents of the whole screen.

\d Discard the part of the input before the cursor.

\wN Advance clock N/10 seconds, possibly causing the current and subsequent inputs to timeout.

\w Advance clock by the amount of real time since this input started (times the current speed factor).

\t Advance clock just enough to timeout the current input

Reverse-Video Display Method Settings

\rn none

\rc CAPS

\rd doublestrike

\ru underline

Output Compression Settings

\cn none: show whole screen before every input.

\cm max: show only lines that have new nonblank characters.

\cs spans: like max, but emit a blank line between each span of screen lines shown.

\chN Hide top N lines (orthogonal to above modes).

Misc Settings

\sfX Set speed factor to X. (0 = never timeout automatically).

\mp Toggle use of MORE prompts

\ln Toggle display of line numbers.

\lt Toggle display of the line type identification chars.

\vb Toggle visual bell.

\pb Toggle display of picture outline boxes.

(Toggle commands can be followed by a 1 or 0 to set value ON or OFF.)

Character Escapes

\\	backslash
\#	backspace
\[	escape
_	return
\<	cursor-left
\>	cursor-right
\^	cursor-up
\.	cursor-down
\1..\0	f1..f10
\D..\X	Standard Frotz hotkeys.

use \H (help) to see the list of hotkeys.

Line Type Identification Characters

Input lines (untimed)

>	A regular line-oriented input
)	A single-character input
}	A line input with some input before the cursor. Use \d to discard it.

Input lines (timed)

T	A regular line-oriented input
t	A single-character input
D	A line input with some input before the cursor. Use \d to discard it.

Output lines

]	Output line that contains the cursor.
.	A blank line emitted as part of span compression.
	(blank) Any other output line.

UNICODE

Dumb Frotz supports Unicode glyphs by way of UTF-8 if the terminal used supports UTF-8. If you prefer using xterm, start it as uxterm. This is a wrapper script that sets up xterm with UTF-8 locale. You can also manually tell an xterm to switch into UTF-8 mode by holding CTRL and the right mouse button to bring up the VT FONTS menu. Depending on how xterm was installed, you may see an option for "UTF-8 Fonts" which will allow Unicode to be properly displayed.

Getting normal xterm to behave like this all the time can vary from system to system. Other terminal emulators have their own ways of being set to use UTF-8 character encoding.

If you are using some sort of wrapper around **dfrotz,** and want to make use of Unicode, it is the responsibility of the wrapper to make sense of characters passed back and forth.

NON ASCII CHARACTERS

Non-ASCII glyphs can be displayed without the use of UTF-8 by way of the ISO-8859-1 or ISO-8859-15 (Latin-1 or Latin-9) character sets. ISO-8859-15 is more or less identical to ISO-8859-1 except that the

OE/oe dipthongs are supported, replacing the seldom-used 1/2 and 1/4 glyphs. See also **luit(1) charsets(7) iso_8859-1(7)** and **iso_8859-15(7)** for more information.

LOCALE

An important means of ensuring the system knows to use UTF-8 is to make sure the locale is set appropriately. This is valid only when **Dumb Frotz** runs under Unix-ish systems.

Using the command **locale** will tell you what is currently in use. Using **locale -a**

will show you what's available. Then set your LANG evironmental variable to something appropriate by using one of these commands:

```
export LANG=C.UTF-8
export LANG=en_US.utf8
```

This can be put in your shell configuration file, be it **.profile, .bash_profile, .login, .bashrc,** or whatever.

It can also be set system-wide in the equivalent files in **/etc.**

SEE ALSO

ash(1) bash(1) csh(1) ksh(1) sh(1) zsh(1)

ENVIRONMENT

Unlike it's curses-using or SDL-using siblings, **dfrotz** does not search any path for game files.

FURTHER INFORMATION

The Frotz homepage is at https://661.org/proj/if/frotz/.

A **git(1)** repository of all versions of Unix Frotz back to 2.32 is available for public perusal here:
https://gitlab.com/DavidGriffith/frotz/.

The bleeding edge of Frotz development may be followed there.

Source tarballs are available at the IF Archive or any of its many mirrors:
http://www.ifarchive.org/

Most distributions of Linux and BSD include Frotz in their package repositories.
It is distributed under the GNU General Public License version 2 or (at your option) any later version.
https://www.gnu.org/licenses/gpl-2.0.en.html
This software is offered as-is with no warranty or liability. If you find a bug or would like **Frotz** to do something it doesn't currently do, please visit the above Gitlab website and report your concerns.

CAVEATS

The Z Machine itself has trouble with the concept of resizing a terminal. It assumes that once the text height and width are set, they will never change; even across saves. This made sense when 24x80 terminals were the norm and graphical user interfaces were mostly unknown. I'm fairly sure there's a way around this problem, but for now, don't resize an xterm in which frotz is running. Also, you should try to make sure the terminal on which you restore a saved game has the same dimensions as the one on which you saved the game.

This manpage is not intended to tell users HOW to play interactive fiction. Refer to the file

HOW_TO_PLAY included in the Unix Frotz documentation or visit one of the following sites:
http://www.microheaven.com/ifguide/
http://www.brasslantern.org/beginners/
http://www.musicwords.net/if/how_to_play.htm
http://ifarchive.org/

BUGS

This program has no bugs. no bugs. no bugs. no *WHAP* thank you. If you find one, please report it to the Gitlab site referenced above in **FURTHER INFORMATION.**

AUTHORS

Frotz was written by Stefan Jokisch for MSDOS in 1995-7.
The Unix port was done by Galen Hazelwood.
The Unix port is currently maintained by David Griffith <dave@661.org>.

SEE ALSO

frotz(6) **sfrotz**(6) **nitfol**(6) **rezrov**(6) **jzip**(6) **xzip**(6) **inform**(1)