EEE - Environment Embedding Executable

Table of Contents

EEE

Environment Embedding Executable

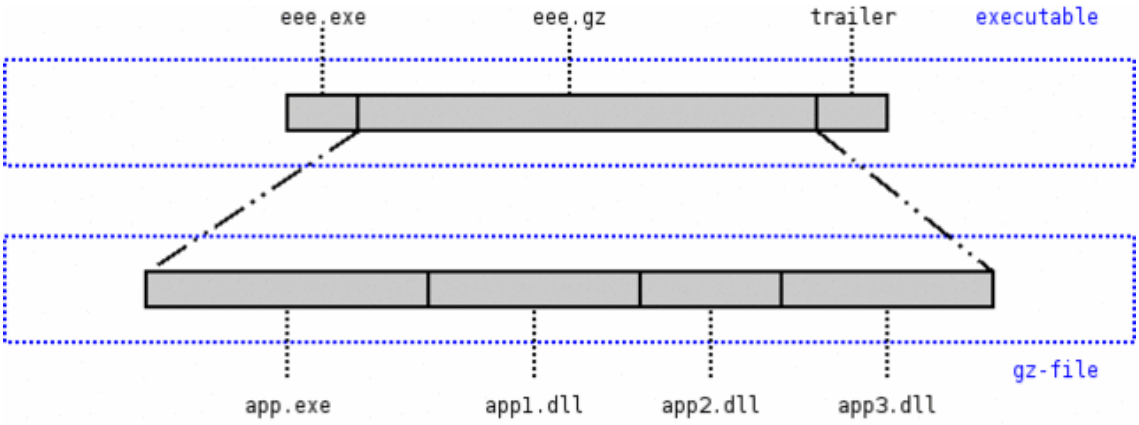
Tue Jul 17 23:37:14 UTC 2007

Erik Veenstra <eee@erikveen.dds.nl>

1. Introduction

EEE stands for "Environment Embedding Executable". Well, I just had to give it a name...

EEE is a program that combines multiple files and actions into a compressed Windows, Linux or Mac OS X (Darwin) executable. Do you have to distribute DLL's or SO's along with your application? The interpreter with you script? The runtime environment with your program? EEE is what you're looking for!



It should be possible to change the icon with a resource editor like Resource Hacker.

2. Internals

EEE is written in Pascal. I use FreePascal, (1.9.8 on Windows, 1.9.8 on Linux and 1.9.8 on Darwin).

EEE has two modes: packing and unpacking. When it detects an attached archive, it jumps into unpacking mode; into packing mode otherwise. Unpacking mode actually means running the application.

2.1. Packing Mode

In packing mode, EEE reads the `eee`-file and appends header and data to a copy of itself for every line it encounters. If a line starts with an `f`, it assumes that the rest of the line is the name of a file in the local directory and attaches the contents of that file as data. A line starting with an `r` packs the given directory and its contents, recursively. Lines starting with either a `t` or `c` are just the actions that have to be performed when running the application. They mean nothing during this phase, EEE just generates a header for each. Lines starting with a `d` or `i` are also disregarded in this phase (they respectively create an empty directory and an INFO-file when unpacking).

2.2. Unpacking Mode / Running the Application

When running the application, EEE generates a temporary directory (`%TEMP%\eee.`*`pid`*) and parses the headers it includes, chronologically: `f`-headers result in files in the tempdir, `r`-headers result in a directory tree in the tempdir, `d`-headers result in an empty directory in the tempdir, `i`-headers result in an INFO-file in the tempdir, `t`-actions are performed in the tempdir and `c`-actions are performed in the directory in which the user started the application.

Every header is scanned for "variables" (lowercase!): `%tempdir%` is replaced by the temporary directory EEE created for dropping its files, `%tempdir1%` is `%tempdir%` with forward slashes, `%tempdir2%` is `%tempdir%` with back slashes, `%orgdir%` is replaced by the directory in which the user started the application, `%parms%` is replaced by the parameters the user has given on the command line and `%quotedparms%` is replaced by the parameters the user has given on the command line, surrounded by single quotes (`'`).

After these actions have finished, EEE starts cleaning up.

The return code of the the the last `t`-action or `c`-action is the return code EEE returns to its caller. (See Return Code.)

3. Usage

See also RubyScript2Exe.

3.1. Quick Reference

a) EEE-file

- *f-lines* tell EEE which *f*iles are to be included.
- *d-lines* tell EEE which *d*irectories are to be created, empty.
- *r-lines* tell EEE which directories are to be packed, *r*ecursively.
- *t-lines* perform an action in the *t*emporary directory.
- *c-lines* perform an action (*c*ommand) in the user directory.
- *i-lines* tell EEE to create an INFO-file in the temporary directory EEE created for dropping its files.

- *%tempdir%* is replaced by the temporary directory EEE created for dropping its files.
- *%parms%* is replaced by the parameters the user has given on the command line.
- *%quotedparms%* is replaced by the parameters the user has given on the command line, surrounded by single quotes (').

b) Packing

```
c:\home\erik> eee eee-file exe-file
```

... or ...

```
c:\home\erik> eeew eee-file exe-file
```

c) Unpacking

To run the application:

```
c:\home\erik> application.exe
```

If you just want to list the contents of the executable, use: (Doesn't work in combination with `eeew`.)

```
c:\home\erik> application.exe --eee-list
```

If you just want to show the information stored in the executable, use: (Doesn't work in combination with `eeew`.)

```
c:\home\erik> application.exe --eee-info
```

If you just want to extract the original files from the executable into the current directory, use:

```
c:\home\erik> application.exe --eee-justextract
```

3.2. INFO-file

EEE creates an INFO-file when it runs in unpacking mode and encounters an `i`-line:

```
i filename
```

An INFO-file contains some information about EEE and the way it is invoked:

```
EEE_APPEXE=filename of the generated executable
EEE_EEEEXE=eee.exe or eeew.exe or eee_linux or eee_darwin
EEE_TEMPDIR=temporary directory in which the application resides
EEE_PARMS=parameters from the command line
EEE_QUOTEDPARMS=quoted parameters from the command line
```

This information can be read by the embedded application, if necessary.

See INFO-file.

4. Tutorial

4.1. Day One

Create a `bat`-file (`test.bat`):

```
@echo off
type %1
```

Create a `txt`-file (`test1.txt`):

```
Hello
```

Create an `eee`-file (`test.eee`), which tells EEE which files to include and what to do with them:

```
f test.bat
f test1.txt

t test.bat test1.txt
```

Pack it:

```
c:\home\erik> eee test.eee test.exe
```

Test it:

```
c:\home\erik> test
Hello
```

4.2. Day Two

Add a line to the `eee`-file of day one:

```
f test.bat
f test1.txt

t test.bat test1.txt
c %tempdir%\test.bat test2.txt
```

Pack it:

```
c:\home\erik> eee test.eee test.exe
```

Create a `txt`-file (`test2.txt`):

```
World!
```

Test it:

```
c:\home\erik> test
Hello
```

```
World!
```

4.3. Day Three

Create a `bat`-file (`test.bat`):

```
@echo off
echo %1 %2
```

Create an `eee`-file (`test.eee`):

```
f test.bat

t test.bat %parms%
t test.bat %quotedparms%
```

Pack it and test it:

```
c:\home\erik> eee test.eee test.exe

c:\home\erik> test Erik Veenstra
Erik Veenstra
'Erik' 'Veenstra'

c:\home\erik> test "Erik Veenstra"
Erik Veenstra
'Erik Veenstra'
```

4.4. Day Four

Create an `eee`-file (`test.eee`):

```
t "C:\Program Files\Internet Explorer\IEXPLORE.EXE" "http://some.domain.tld/"
```

(Or whatever location IE resides in...)

Pack it:

```
c:\home\erik> eee test.eee test.exe
```

Double-click on the `test.exe`. A DOS-box will appear.

Again, but with `eeew.exe`:

```
c:\home\erik> eeew test.eee test.exe
```

Double-click on the `test.exe`. No DOS-box.

5. Examples

5.1. RubyScript2Exe

EEE is one of the main parts, if not *the* main part, of RubyScript2Exe.

5.2. AllInOneRuby

It's sometimes not easy, or possible, or desirable, or allowed to do a complete Ruby installation. That's where AllInOneRuby comes in. I always have a USB-memory stick with AllInOneRuby in my pocket.

5.3. AllInOneQEMU

I created AllInOneQEMU to able to run my own Linux environment on top of a Windows machine, straight from USB. No installation, no garbage. AllInOneQEMU is part of QEMU-Puppy.

5.4. AWK

I regularly use EEE for packing and distributing AWK scripts. It's amazing that a lot of colleagues (IT-professionals !) can't remember "awk -f *script*.awk"...

```
f awk.exe
f script.awk

c %tempdir%\awk.exe -f %tempdir%\script.awk %parms%
```

Then "compile" it with:

```
eee.exe script.eee script.exe
```

5.5. INFO-file

```
$ cat test.eee
i infofile.txt
t cat infofile.txt

$ ./eee_linux test.eee test_linux

$ ./test_linux a 'b c'
EEE_APPEXE=/home/erik/scratch/test_linux
EEE_EEEEXE=eee_linux
EEE_TEMPDIR=/tmp/eee.580
EEE_PARMS=a b c
EEE_QUOTEDPARMS='a' 'b c'
```

5.6. Return Code

```
$ cat test.eee
c exit 17

$ ./eee_linux test.eee test_linux

$ ./test_linux

$ echo $?
17
```

6. License

6.1. License of EEE

EEE, Copyright (C) 2003 Erik Veenstra <eee@erikveen.dds.nl>

This program is free software; you can redistribute it and/or modify it under the terms of the *GNU General Public License (GPL), version 2*, as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, **but without any warranty**; without even the implied warranty of **merchantability** or **fitness for a particular purpose**. See the *GNU General Public License (GPL)* for more details.

You should have received a copy of the *GNU General Public License (GPL)* along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

The full text of the license can be found here.

6.2. License of your Application

Whatever...

7. Download

EEE is developed as a generic and reusable part of RubyScript2Exe. RubyScript2Exe has releases and versions. EEE itself doesn't. It's just there, always the newest, even newer than the one in the last release of RubyScript2Exe. Nevertheless, it's stable.

Tested on:

- Windows 95
- Windows 98
- Windows 2000
- Windows XP
- Linux (IA32)
- Linux (PPC)
- Darwin

Download one of the binaries `eee.exe` (Windows, with a DOS-box), `eeew.exe` (Windows, without a DOS-box), `eee_linux` (Linux, libc >=2.3.2) or `eee_darwin` (Darwin).
Or download the source `eee.pas` and compile it yourself. (On Windows, you also need `eee.rc` and an icon with the name `eee.ico`, like `the one for RubyScript2Exe`.)

It's compiled with FreePascal, (1.9.8 on Windows, 1.9.8 on Linux and 1.9.8 on Darwin):

```
c:\home\erik> copy eee.pas eeew.pas
c:\home\erik> windres -i eee.rc -o eee.res
c:\home\erik> fpc -Xs -B eee
c:\home\erik> fpc -Xs -B -WG eeew
```

Send me reports of all bugs and glitches you find. Propositions for enhancements are welcome, too. This helps *us* to make *our* software better.

8. Known Issues

None...