AllInOneRuby - A "Just-in-Time and Temporary Installation of Ruby"

Table of Contents

AllInOneRuby

A "Just-in-Time and Temporary Installation of Ruby"

Fri May 25 15:05:24 UTC 2007

Erik Veenstra <allinoneruby@erikveen.dds.nl>

1. Introduction

As you know, Ruby has to be installed, either by unzipping a ZIP-file, or by running an installer. Only then, you can run a script. Well, not anymore!

AllInOneRuby creates a compressed executable for Windows, Linux or Mac OS X (Darwin) that includes both the Ruby interpreter and the runtime libraries. Why? Because 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.

A "just-in-time and temporary installation of Ruby"...

Because of the gathering of files from your own Ruby installation, AllInOneRuby creates an executable for the platform it's being run on. No cross compile.

And when I say Windows, I mean both Windows (RubyInstaller, MinGW and MSWin32) and Cygwin. But the generated `exe` under Cygwin is very, very big, because its `exe`'s are very big (static?) and it includes `cygwin1.dll`, so it can run on machines without Cygwin.

What's the difference between RubyScript2Exe and AllInOneRuby? Well, RubyScript2Exe includes an application (your script), the Ruby VM and only parts of the rubylib tree (it's stripped specifically for that application). AllInOneRuby contains a complete Ruby installation: it includes no application, but it does include the Ruby VM and the complete rubylib tree. You can use `allinoneruby.exe` like `ruby.exe` (Windows) and `allinoneruby_*` like `ruby` (Linux, Darwin) that's already installed on your system. In other words: the executable, generated with RubyScript2Exe, is an application; the one generated with AllInOneRuby "is" Ruby.

2. Internals

AllInOneRuby runs in 3 phases:

1. The files `ruby.exe`, `rubyw.exe` and their `dll`'s (determined recursively) in the directory `Config::CONFIG["bindir"]` are copied to the directory `bin/`.
2. All subdirectories in the directory `Config::CONFIG["libdir"]` that are part of `Config::CONFIG["rubylibdir"]` are recursively copied to the directory `lib/`. The `dll`'s (determined recursively in the directory `Config::CONFIG["bindir"]`) of the `so`'s it encounters are copied to `bin`/.
3. Finally, these directories (`bin/` and `lib/`) are compressed and packed with EEE. EEE was originally written as part of RubyScript2Exe.

That's it.

The files in `Config::CONFIG["sitelibdir"]` are completely ignored, because (strictly spoken) they are not part of Ruby. Unless you use `-s` or `--site` .

3. Usage

3.1. AllInOneRuby

If you use the original `allinoneruby.rb`, use this to create *`allinoneruby`*`.exe`:

```
c:\home\erik> ruby allinoneruby.rb [allinoneruby.exe] [parameters]
```

If you installed the gem, it's:

```
c:\home\erik> allinoneruby [allinoneruby.exe] [parameters]
```

Parameter	Description
`--rubyw` *or* `-w`	Avoid the popping up of a DOS box.
`--ruby` *or* `-d`	Force the popping up of a DOS box (default).
`--site` *or* `-s`	Include all files in `Config::CONFIG["sitelibdir"]` as well.
`--nostrip` *or* `-ns`	Avoid stripping. The binaries (`ruby` and `*.so`) on Linux and Darwin are stripped by default to reduce the size of the resulting executable.
`--verbose` *or* `-v`	Verbose mode.
`--quiet` *or* `-q`	Quiet mode.

On Linux and Darwin, there's no difference between `ruby` and `rubyw`.

3.2. The Executable (EEE)

To run *`allinoneruby`*`.exe`:

```
c:\home\erik> allinoneruby.exe script.rb [parameters]
or
c:\home\erik> allinoneruby.exe -e 'statements' [parameters]
```

Parameter	Description
`--eee-list`	Just list the contents of the executable. (Doesn't work in combination with `rubyw`.)
`--eee-info`	Just show the information stored in the executable. (Doesn't work in combination with `rubyw`.)
`--eee-justextract`	Just extract the original files from the executable into the current directory (no subdirectory!).

If one of these parameters is used, AllInOneRuby does just that. It doesn't execute *`script.rb`*.

If none of these parameters is used, AllInOneRuby executes *`script.rb`* with the given parameters. To be forward compatible, all parameters starting with `--eee-` will be deleted before the execution of *`script.rb`*.

The exit code of the executable is the same as the exit code of *`script.rb`*.

a) The Application

The application can detect whether it is run by AllInOneRuby or not with `defined?(ALLINONERUBY)` (which is set to `ruby`, `ruby.exe` or `rubyw.exe`). If the application is not run by AllInOneRuby, this will return `nil`.

In your application, you can access some information about the environment EEE sets up before spawning your application:

Constant	Set to
`ALLINONERUBY_APPEXE`	Filename of the generated executable.
`ALLINONERUBY_EEEEXE`	`eee.exe` or `eeew.exe` or `eee_linux` or `eee_darwin`.
`ALLINONERUBY_TEMPDIR`	Temporary directory in which the application resides.
`ALLINONERUBY_PARMS`	Parameters from the command line.
`ALLINONERUBY_QUOTEDPARMS`	Quoted parameters from the command line.

4. Examples

4.1. AllInOneRuby

```
c:\home\erik> ruby allinoneruby.rb [allinoneruby.exe]
```

This generates `allinoneruby.exe` .

4.2. Distributions

I ran AllInOneRuby with four different Ruby distributions (Ruby 1.8.1) on Windows and two version of Ruby (1.6.7 and 1.8.2) on Linux:

Distribution	Size (bytes)
Cygwin	4830863
RubyInstaller	1575412
MinGW	2660076
MSWin32	1493915
Linux, Ruby 1.6.7	1045474
Linux, Ruby 1.8.2	3435857

5. License

AllInOneRuby, Copyright (C) 2003 Erik Veenstra <allinoneruby@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. Download

Current version is 0.2.11 (15.04.2007). It's a stable release.

Tested on:

- Red Hat Linux 8.0 with Ruby 1.6.7
- Red Hat Linux 8.0 with Ruby 1.8.1
- Red Hat Linux 8.0 with Ruby 1.8.2
- Windows 98 with Ruby 1.8 (RubyInstaller)
- Windows 2000 with Ruby 1.8 (Cygwin)
- Windows 2000 with Ruby 1.8 (RubyInstaller)
- Windows 2000 with Ruby 1.8 (MinGW)
- Windows 2000 with Ruby 1.8 (MSWin32)
- Windows XP with Ruby 1.8 (Cygwin)
- Windows XP with Ruby 1.8 (RubyInstaller)
- Windows XP with Ruby 1.8 (MinGW)
- Windows XP with Ruby 1.8 (MSWin32)

You only need allinoneruby.rb . It's the current version, packed as an RBA (Ruby Archive, built by Tar2RubyScript) and works on both Windows and Linux. You can download allinoneruby.tar.gz if you want to play with the internals of AllInOneRuby. AllInOneRuby is available as allinoneruby.gem as well.

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

A change log and older versions can be found here. A generated log file can be found here.

AllInOneRuby is available on SourceForge.net and on RubyForge .

6.1. Mac OS X (Darwin)

I included (experimental) support for Darwin. The Ruby code in the above mentioned packages is able to handle Darwin, but the packages don't include EEE for Darwin. (They would be too big...) For now, you have to compile it yourself:

1. Get `eee.pas` from the RubyScript2exe archive.
2. Download the compiler.
3. Compile (`fpc -Xs -B eee.pas`).
4. Rename `eee` to `eee_darwin`.

(I've put a precompiled `eee_darwin` on my site, but it may be newer than (and therefor incompatible with) the released Ruby code.)

AllInOneRuby searches for `eee_darwin` (or `eee_linux` or `eee.exe` or `eeew.exe`) in 3 locations:

6. Download

1. In `allinoneruby.rb` (or `allinoneruby/` when you are using `allinoneruby.tar.gz`).
2. In the directory in which `allinoneruby.rb` is located.
3. In the current directory.

This means that you can simply put `eee_darwin` in the same directory as `allinoneruby.rb` (location 2) or in the current directory (location 3).

If you want to repackage AllInOneRuby (location 1) with an embedded `eee_darwin`, do this:

1. Extract allinoneruby.tar.gz, or extract allinoneruby.rb (`ruby allinoneruby.rb --tar2rubyscript-justextract`)
2. Copy `eee_darwin` to `allinoneruby/`.
3. Recreate allinoneruby.rb (`ruby tar2rubyscript.rb allinoneruby/`) (optional)

7. Known Issues

- On Darwin, you might run into a `Too many open files - getcwd (Errno::EMFILE)`. I've no solution for this. As far as I know, this is not directly related to AllInOneRuby or Tar2Rubyscript. It's an OS kind of thing. Run ulimit -n to show the maximum number of open files. It is 256 on Mac OS X. Run ulimit -n3000 (for example) to fix it.
- On Windows, `if __FILE__ == $0` doesn't work. `__FILE__` is set to ./*script*.rb instead of *script*.rb. **FIXED IN 0.2.7!**
- If you someday run into trouble when trying to generate RDOC documentation from within the generated executable, have a look at Zoran Lazarevic's page.
- Because of the way Ruby is started from within AllInOneRuby on Linux, `$stdin` doesn't work anymore. On Windows, it works.
- Running the generated executable from within a non-SH-compatible shell (e.g. TCSH), is currently not possible. Will be fixed in the next release.