

## Ruby master - Feature #6418

### Supporting a subset of ANSI escape code on Windows

05/09/2012 11:04 AM - usa (Usaku NAKAMURA)

<b>Status:</b>	Closed
<b>Priority:</b>	Normal
<b>Assignee:</b>	usa (Usaku NAKAMURA)
<b>Target version:</b>	2.0.0
<b>Description</b>	
Are there those who need this?	

#### Associated revisions

##### Revision 82aa4340 - 06/04/2012 06:39 AM - nobu (Nobuyoshi Nakada)

win32: VT100 escape

- win32/win32.c (constat\_apply): apply VT100 functions. [ruby-core:44958] [Feature #6418]
- win32/win32.c (constat\_parse): parse some VT100 escape sequence.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35907 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 35907 - 06/04/2012 06:39 AM - nobu (Nobuyoshi Nakada)

win32: VT100 escape

- win32/win32.c (constat\_apply): apply VT100 functions. [ruby-core:44958] [Feature #6418]
- win32/win32.c (constat\_parse): parse some VT100 escape sequence.

##### Revision 35907 - 06/04/2012 06:39 AM - nobu (Nobuyoshi Nakada)

win32: VT100 escape

- win32/win32.c (constat\_apply): apply VT100 functions. [ruby-core:44958] [Feature #6418]
- win32/win32.c (constat\_parse): parse some VT100 escape sequence.

##### Revision 35907 - 06/04/2012 06:39 AM - nobu (Nobuyoshi Nakada)

win32: VT100 escape

- win32/win32.c (constat\_apply): apply VT100 functions. [ruby-core:44958] [Feature #6418]
- win32/win32.c (constat\_parse): parse some VT100 escape sequence.

##### Revision 35907 - 06/04/2012 06:39 AM - nobu (Nobuyoshi Nakada)

win32: VT100 escape

- win32/win32.c (constat\_apply): apply VT100 functions. [ruby-core:44958] [Feature #6418]
- win32/win32.c (constat\_parse): parse some VT100 escape sequence.

##### Revision 35907 - 06/04/2012 06:39 AM - nobu (Nobuyoshi Nakada)

win32: VT100 escape

- win32/win32.c (constat\_apply): apply VT100 functions. [ruby-core:44958] [Feature #6418]
- win32/win32.c (constat\_parse): parse some VT100 escape sequence.

##### Revision 35907 - 06/04/2012 06:39 AM - nobu (Nobuyoshi Nakada)

win32: VT100 escape

- win32/win32.c (constat\_apply): apply VT100 functions. [ruby-core:44958] [Feature #6418]
- win32/win32.c (constat\_parse): parse some VT100 escape sequence.

#### History

##### #1 - 05/09/2012 11:16 AM - drbrain (Eric Hodel)

RDoc uses ANSI escape sequences for ri output.

**#2 - 05/09/2012 11:18 AM - drbrain (Eric Hodel)**

Here is an example command that exercises ri ANSI output:

```
./miniruby -I./lib -I. -I.ext/common ./tool/runruby.rb --extout=.ext -- bin/ri --no-standard -d .ext/rdoc Regexp -f ansi
```

**#3 - 05/09/2012 11:37 AM - luislavena (Luis Lavena)**

usa (Usaku NAKAMURA) wrote:

Are there those who need this?

A lot of libraries uses ANSI without offering non-colored (non-ANSI) alternatives.

Windows users needed to depend on win32console (back in 1.8x) and now external tools like ANSICON to workaround that but facing side-effects of it.

Having a basic subset of ANSI supported on Windows will be really helpful to reduce such issues.

What you see as drawback from this approach? It is considering redirected handles? (so ANSI is written properly?)

Thank you.

**#4 - 05/09/2012 06:29 PM - usa (Usaku NAKAMURA)**

- File *ansi\_escape2.diff* added

(updated the patch)

**#5 - 05/10/2012 02:30 AM - vo.x (Vit Ondruch)**

Seems like case for extension gem. But we'd been there already and it was failure. Not sure why this should be reintroduced and even directly into the Ruby. Let's keep focus on correct Ruby functionality and ANSI escape characters leave to terminals.

**#6 - 05/10/2012 02:40 AM - luislavena (Luis Lavena)**

vo.x (Vit Ondruch) wrote:

Seems like case for extension gem. But we'd been there already and it was failure. Not sure why this should be reintroduced and even directly into the Ruby. Let's keep focus on correct Ruby functionality and ANSI escape characters leave to terminals.

There is no single terminal on Windows that properly captures ANSI codes from executing programs. Terminal emulations like mintty do not work properly when mixing file descriptor between native Windows and posix-emulated ones (msys/cygwin)

External tools like ANSICON while solved the coloring issue introduced other issues, like hooking up Command Processor AutoRun which resulted in failure to spawn child processes or commands that dependent on cmd.exe

There is no easy way to provide this as an extension since the particular function that provides this is an internal part of Ruby replacement of IO management on Windows.

While working with RubyInstaller I've received several complains about these issues, the old win32console issues too and from time to time comparison with Node JS out-of-the-box ANSI coloring in the default terminal (cmd.exe)

I haven't had the time to analyze the impact (e.g. performance) on this, but based on previous experience with external tools and gems, I do think the impact on having this out-of-the-box is valuable.

In relation to "focus on correct Ruby functionality" I've worked on a rewrite of File.expand\_path and the internal counterpart that cut Ruby startup time half the current versions.

Does that qualifies for correct Ruby functionality?

Thank you.

**#7 - 05/10/2012 07:12 PM - nobu (Nobuyoshi Nakada)**

- File *ansi\_escape3.diff* added

=begin

Some improvements.

- better multi-params sequence
- bold by intensity

- allow split output =end

#### #8 - 05/11/2012 07:44 AM - luislavena (Luis Lavena)

nobu (Nobuyoshi Nakada) wrote:

Some improvements.

Thank you Nobu,

I've run some naive tests against it, simulating a big and fast test suite and using minitest/pride to report colorful output:

<https://gist.github.com/2655794>

The tests were performed from a RAMdisk where Ruby and the dependencies were installed. The numbers were taken after several runs (warm up cache)

From the report:

DEFAULT - No Pride or ANSI coloring was enabled.

PRIDE - minitest/pride enabled, causing ANSI sequences be shown in the terminal

ANSICON - pride enabled and running from a second (child) cmd.exe with ANSICON performing the ANSI processing

ANSI with patch - the latest patch provided by Nobu.

Results of both normal and redirected output were measured.

#### #9 - 05/11/2012 12:00 PM - nobu (Nobuyoshi Nakada)

- File *ansi\_escape3.diff* added

Recent patch:

<https://github.com/ruby/ruby/pull/125>

#### #10 - 06/03/2012 12:22 AM - luislavena (Luis Lavena)

Nobu, Usa,

Apply plans to apply this?

I've been using it locally with good results. Tools like Spinach, MiniTest and even Bundler worked as expected (with colors)

Here I did a naive attempt to benchmark the impact of this change:

<https://gist.github.com/2655794>

And seems the overhead is way lower than using external tools like ANSICON.

Thank you.

#### #11 - 06/04/2012 03:39 PM - nobu (Nobuyoshi Nakada)

- Status changed from *Assigned* to *Closed*

- % Done changed from 0 to 100

This issue was solved with changeset r35907.

Usaku, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

---

win32: VT100 escape

- win32/win32.c (constat\_apply): apply VT100 functions. [ruby-core:44958] [Feature #6418]
- win32/win32.c (constat\_parse): parse some VT100 escape sequence.

#### Files

---

ansi_escape.diff	6.1 KB	05/09/2012	usa (Usaku NAKAMURA)
ansi_escape2.diff	6.54 KB	05/09/2012	usa (Usaku NAKAMURA)
ansi_escape3.diff	9.14 KB	05/10/2012	nobu (Nobuyoshi Nakada)
ansi_escape3.diff	9.14 KB	05/11/2012	nobu (Nobuyoshi Nakada)