

Redmine - Defect #1682

Show error if SCM executable is not in rails PATH (ENV['PATH'])

2008-07-24 21:09 - Adam Grant

Status:	Closed	Start date:	2008-07-24
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	SCM	Estimated time:	0.00 hour
Target version:	0.8	Affected version:	
Resolution:	Fixed		

Description

Redmine should log a message if the scm executable cannot be found in the user's path that the redmine app is running under.

For our case, the svn command was not in the user's PATH, but there was no way to figure this out without custom loggers and 2 hours of debugging.

The subversion_adapter.rb tries to shellout the command "svn list..." and "svn info..." for gathering info on the connected repository. After it tries this, IO.popen creates a variable called \$? which has info on the pipe it created. Specifically, it has an exitstatus, which tells you if the command was successful, or any error it encountered.

Line 105 of lib/redmine/scm/adapters/subversion_adapter.rb has this:

```
return nil if $? && $?.exitstatus != 0
```

The \$?.exitstatus for us was coming out to be 127, which is "command not found". No log was being created to warn admins about this though, and everything failed silently. It ignores all the important exitstatus numbers (besides 0) and doesn't report any problems whatsoever.

It should put a log message like:

```
logger.info "Your SCM (#{your_scm_command_here}) was not found in the PATH (#{ENV['PATH']})"
```

Or at least should log:

```
logger.debug("SCM Process Status: #{ $? .inspect }")
```

which will show the exit code.

Not sure if this should bubble up to the repository setup page on the site, but it would be nice so you don't have to dig through the log just to figure out the error.

Associated revisions

Revision 1757 - 2008-08-25 14:31 - Jean-Philippe Lang

More detailed error message in log when scm command fails (#1682).

History

#1 - 2008-08-02 00:48 - Adam Grant

- File redmine_patch.txt added

Here's a patch for solving the problem. Not tested though. Seems simple enough...

=> lib/redmine/scm/adapters/abstract_adapter.rb

```
Index: abstract_adapter.rb
--- abstract_adapter.rb (revision 1694)
+++ abstract_adapter.rb (working copy)
@ -176,6 +176,9 @
io.close_write
block.call(io) if block_given?
end
```

```
if $? && $??.exitstatus != 0
+   raise Errno::ENOENT, "Exit status is #{ $??.exitstatus }"
+   end
rescue Errno::ENOENT => e
msg = strip_credential(e.message)      # The command failed, log it and re-raise
```

#2 - 2008-08-14 01:37 - Adam Grant

Does the Issues table not actually show the correct updated_at time? I've updated this thread twice, and it still shows up as being updated on the creation date...

#3 - 2008-08-14 14:46 - Thomas Lecavelier

It looks like it's a bug from this instance of redmine: mine (@r1682) has not that issue.

#4 - 2008-08-25 14:32 - Jean-Philippe Lang

- Status changed from New to Closed
- Target version set to 0.8
- Resolution set to Fixed

Error message changed in [r1757](#).

Errno::ENOENT should be automatically raised if the binary is not available. Wasn't it the case for you? If not, what's your OS? Please reopen if needed.

#5 - 2008-08-25 14:48 - Jean-Philippe Lang

Adam Grant wrote:

Does the Issues table not actually show the correct updated_at time? I've updated this thread twice, and it still shows up as being updated on the creation date...

Fixed in [r1758](#) (but not yet applied here). It was a side effect of Rails 2.1 "Dirty objects" feature.

Files

redmine_patch.txt	555 Bytes	2008-08-01	Adam Grant
-------------------	-----------	------------	------------