I'm using a plugin in a Ruby on Rails application that uses FileUtils's cp_r method to copy files into place (with verbose mode on).

When deploying my application with Capistrano, I get several "error" lines during this part (http://groups.google.com/group/capistrano/browse_thread/thread/64e95a26af3035cd?pli=1).

It appears that FileUtils#cp_r prints all output in verbose mode to stderr, even if the output is not erroneous (http://stackoverflow.com/questions/3163585/getting-executed-command-from-ruby-fileutils). It seems that fu_output_message (and related code) should be changed so that output goes to stdout and errors go to stderr, as usual.

This was set to the Ruby 1.8 tracker, but I think it is still an issue currently for all FileUtils methods. Changing the default behavior could possibly be done in Ruby 3, and potentially we could add an option for $stdout (or another IO stream) for non-error messages in 2.7.

After a more thorough review, fu_output_message is only called in verbose mode. You can get the behavior you want via:

FileUtils.instance_variable_set(:@fileutils_output, $stdout)

before calling other methods on FileUtils. I think changing the default from $stderr to $stdout should be done in Ruby 3.

I've submitted a pull request upstream for this: https://github.com/ruby/fileutils/pull/49