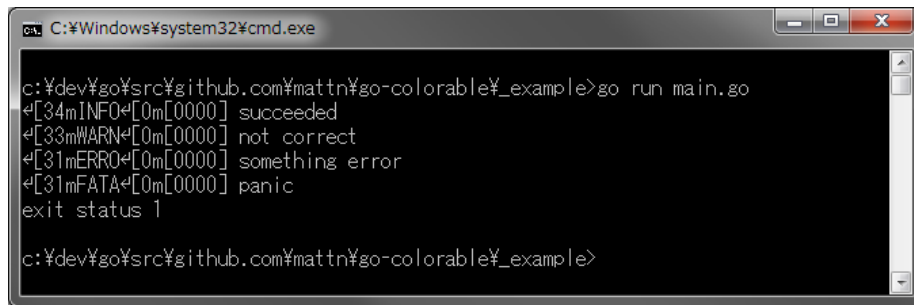


go-colorable

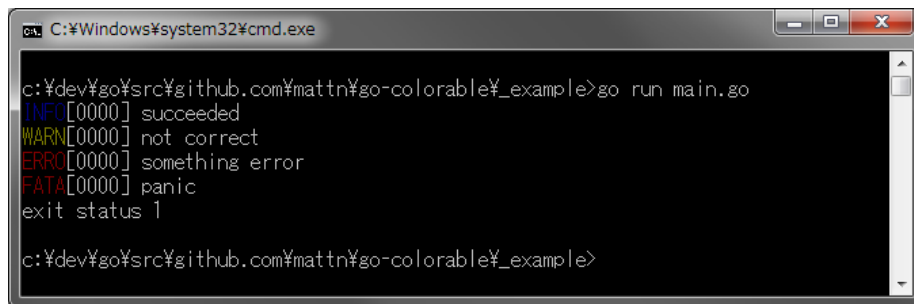
Colorable writer for windows.

For example, most of logger packages doesn't show colors on windows. (I know we can do it with ansicon. But I don't want.) This package is possible to handle escape sequence for ansi color on windows.

Too Bad!

A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The prompt shows the command `c:\dev\go\src\github.com\mattn\go-colorable_example>go run main.go`. The output is as follows:
`✓[34mINFO✓[0m[0000] succeeded
✓[33mWARN✓[0m[0000] not correct
✓[31mERRO✓[0m[0000] something error
✓[31mFATA✓[0m[0000] panic
exit status 1
c:\dev\go\src\github.com\mattn\go-colorable_example>`
The escape sequences (e.g., `✓[34m`) are visible, indicating that the colors are not rendered.

So Good!

A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The prompt shows the command `c:\dev\go\src\github.com\mattn\go-colorable_example>go run main.go`. The output is as follows:
`INFO[0000] succeeded
WARN[0000] not correct
ERRO[0000] something error
FATA[0000] panic
exit status 1
c:\dev\go\src\github.com\mattn\go-colorable_example>`
The log levels and messages are displayed in their respective colors (blue for INFO, yellow for WARN, red for ERRO and FATA), indicating that the colors are rendered correctly.

Usage

```
logrus.SetFormatter(&logrus.TextFormatter{ForceColors: true})  
logrus.SetOutput(colorable.NewColorableStdout())  
  
logrus.Info("succeeded")  
logrus.Warn("not correct")  
logrus.Error("something error")  
logrus.Fatal("panic")
```

You can compile above code on non-windows OSs.

Installation

```
$ go get github.com/mattn/go-colorable
```

License

MIT

Author

Yasuhiro Matsumoto (a.k.a mattn)