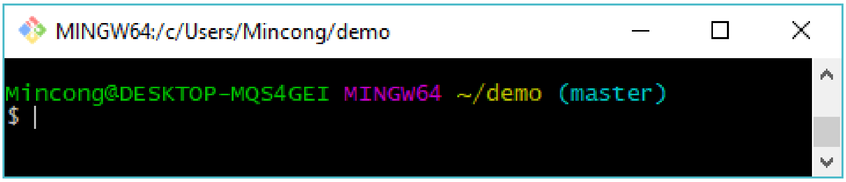
Customize Git Prompt in MinGW

Overview

In this article, we will see how to customize Git prompt in MinGW64 (Minimalist GNU for Windows x64). By default, when current directory is a Git repo, MinGW only show limited information about the repository: the name of the current branch. It does not show the current states, such as dirty-state, untracked files, stash, upstream. This is not practical.



```
Mincong@DESKTOP-MQS4GEI MINGW64 ~/demo (master)
$ |
```

Only branch name *"master"* is shown. More status?

Short Answer

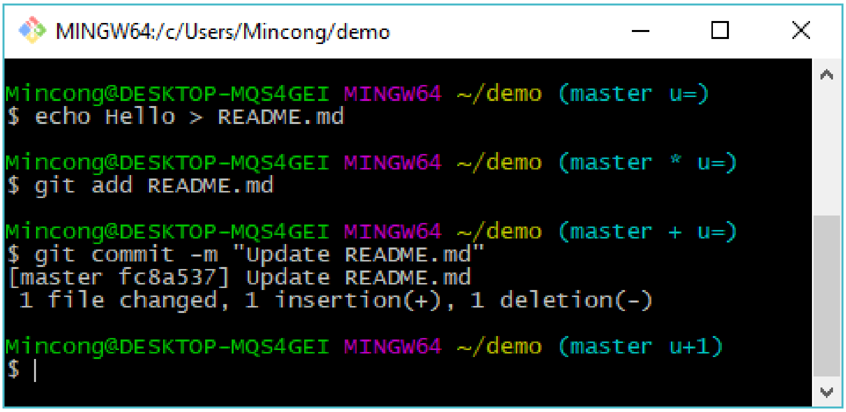
Assume that the MinGW64 (Git Bash console) is installed when you downloaded the Git client from internet. Add these lines into the Bash profile `~/.bash_profile`:

```
GIT_PS1_SHOWDIRTYSTATE=1
GIT_PS1_SHOWUNTRACKEDFILES=1
GIT_PS1_SHOWSTASHSTATE=1
GIT_PS1_SHOWUPSTREAM="auto verbose"
```

and then reload the Bash profile:

```
$ source ~/.bash_profile
```

Now, you'll see more detail about different states in your Git projects, including dirty-state, untracked files, stash and upstream.



```
Mincong@DESKTOP-MQS4GEI MINGW64 ~/demo (master u=)
$ echo Hello > README.md

Mincong@DESKTOP-MQS4GEI MINGW64 ~/demo (master * u=)
$ git add README.md

Mincong@DESKTOP-MQS4GEI MINGW64 ~/demo (master + u=)
$ git commit -m "Update README.md"
[master fc8a537] Update README.md
 1 file changed, 1 insertion(+), 1 deletion(-)

Mincong@DESKTOP-MQS4GEI MINGW64 ~/demo (master u+1)
$ |
```

Long Answer

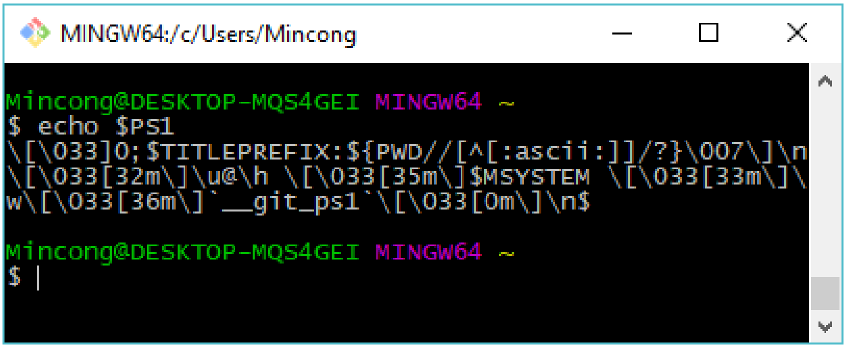
If you want to know more about Git prompt, let's continue for a more detailed answer. In the following paragraphs, I will explain:

- What is PS1, and how it works
- What is Git PS1
- How Git PS1 is linked to PS1

PS1

In MinGW, you can see the value of custom prompt (PS1) by printing variable `$PS1`. And there're two important notions here, they're Bash prompt escape sequences and shell coloring. We will use them to understand the secrets behind PS1 :)

```
$ echo $PS1
\[\033]0;$TITLEPREFIX:${PWD//[^[:ascii:]]/?}\007\]\n\[\033
[32m\]\u@\h \[\033[35m\]$MSYSTEM \[\033[33m\]\w\[\033[36m
\]`__git_ps1`\[\033[0m\]\n$
```



```
Mincong@DESKTOP-MQS4GEI MINGW64 ~
$ echo $PS1
\[\033]0;$TITLEPREFIX:${PWD//[^[:ascii:]]/?}\007\]\n
\[\033[32m\]\u@\h \[\033[35m\]$MSYSTEM \[\033[33m\]\
w\[\033[36m\]`__git_ps1`\[\033[0m\]\n$

Mincong@DESKTOP-MQS4GEI MINGW64 ~
$ |
```

The first line is the header of the window:

- The first line starts with the title prefix of the terminal. In my case, it's *"MINGW64"*. Then, followed by colon (`:`).
- The first line continues with the current directory using PWD (print work directory). Note that the result is filtered by an ASCII filter, where other characters will be displayed as a question mark (`?`).

The second line:

- The second line starts with the username of the current user `\u` , followed by symbol `@` , followed by the hostname up to the first dot (`.`). The color of this section is regular green (`\[\033[32m\]`).
- The second line continues with the system type `$MSYSTEM` (stands for Microsoft system?). This value can

be `MINGW32`, `MINGW64`, or others. The color of this section is regular purple (`\[\033[35m\]`).

- The second line continues with the absolute path of the current working directory `\w` (w in lowercase). Note that w in uppercase `\W` will only show the last segment of the current working directory. The color of this section is regular yellow (`\[\033[33m\]`).
- The second line ends with the Git prompt expression `__git_ps1`. We'll go further on it in the next paragraph. The color of this section is regular cyan (`[\033[36m\]`).

After all, there're still a line feed (`\n`) for starting a new line and a dollar symbol (`$`), which often signifies the end of the Bash prompt and the start of the user command.

Git Prompt (__git_ps1)

The Bash/Zsh Git prompt support is handled by script git-prompt.sh. This script allows you to see repository status in your prompt. You can define your own preferences by providing expressions `GIT_PS1_*` to your terminal. As far as expression `__git_ps1` is called in your PS1 substitution, the Git status will show in your prompt.

In MinGW, we have already seen that `__git_ps1` is called in PS1 substitution:

```
$ echo $PS1
...`__git_ps1`\[\033[0m\]\n$
```

So that's why we can see the Git status.

Conclusion

In this article, we learnt how to customize Git prompt in Bash prompt using expressions `GIT_PS1.*`. It allows us to see more than the current branch name in Bash prompt: dirty-state, untracked files, stash and upstream. We also understand how Bash prompt is displayed via variable `$PS1`; what is Git Prompt and how it is linked to Bash prompt via `__git_ps1`. Hope you enjoy this article, see you the next time!