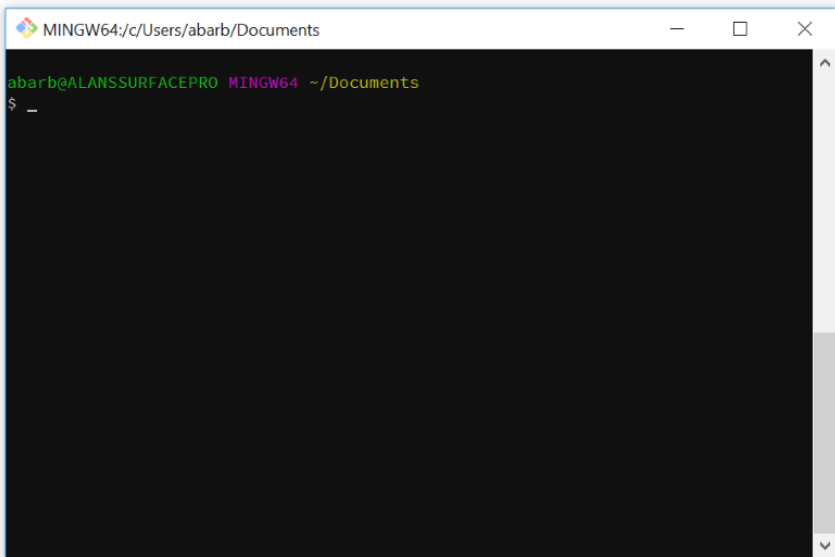


How To Customize The Git For Windows Bash Shell Prompt

Have you upgraded to the latest version of Git for Windows and found the default prompt to be a bit annoying with the new added and useless "MINGW32" or "MINGW64"? Perhaps you don't like how it has the username@hostname or maybe just the color scheme.



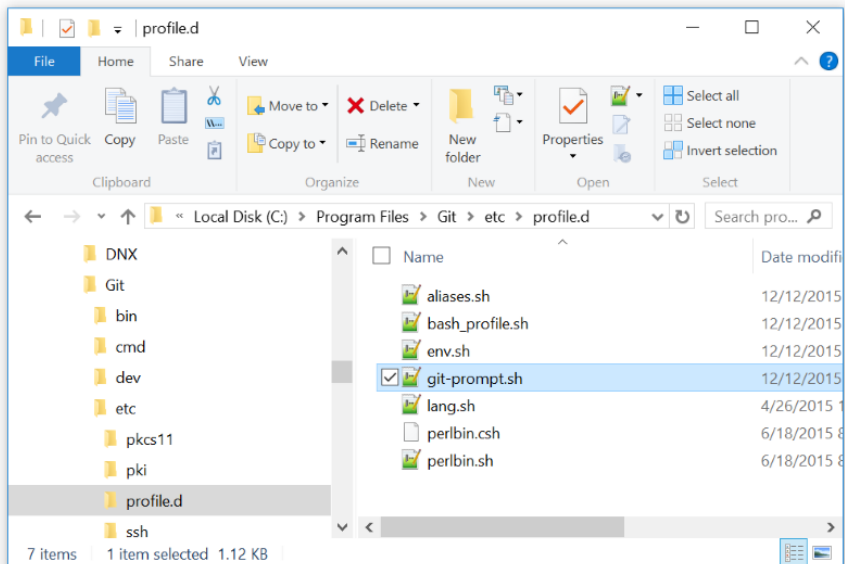
"Git for Windows Bash prompt before customization")

I will show you how you can change the prompt in Git for Windows very easily!

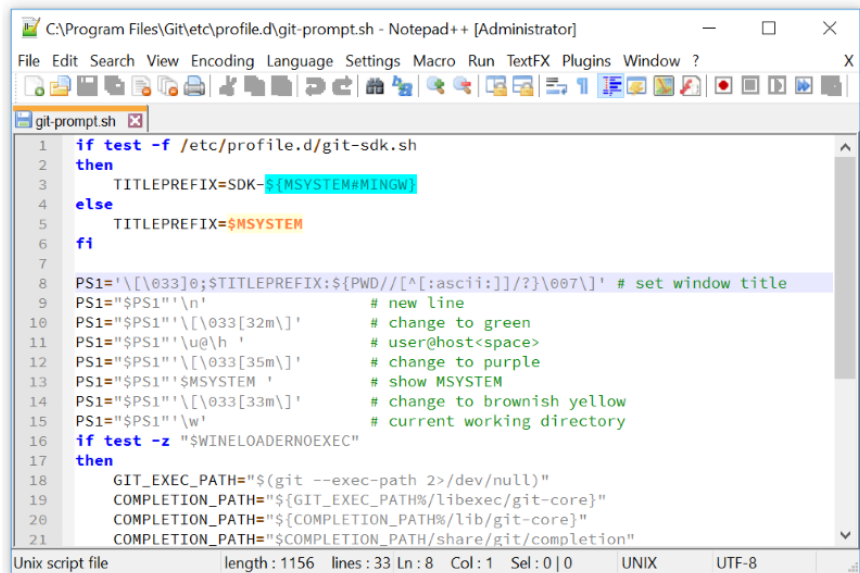
The Git for Windows Bash prompt is set using a configuration shell script named *git-prompt.sh* that can be

found in the following directory.

C:\Program Files\Git\etc\profile.d



Using a text editor that supports unix style files, such as Notepad++, open the file and you will find something that looks like this.



```
1  if test -f /etc/profile.d/git-sdk.sh
2  then
3      TITLEPREFIX=SDK-${MSYSTEM#MINGW}
4  else
5      TITLEPREFIX=$MSYSTEM
6  fi
7
8  PS1='\[\033]0;$TITLEPREFIX:$PWD/[^\[:ascii:]]/?\[\007\]' # set window title
9  PS1="$PS1"'\n' # new line
10 PS1="$PS1"'\[\033[32m\' # change to green
11 PS1="$PS1"'\u@h ' # user@host<space>
12 PS1="$PS1"'\[\033[35m\' # change to purple
13 PS1="$PS1"'$MSYSTEM ' # show MSYSTEM
14 PS1="$PS1"'\[\033[33m\' # change to brownish yellow
15 PS1="$PS1"'\w' # current working directory
16 if test -z "$WINELOADERNOEXEC"
17 then
18     GIT_EXEC_PATH="$(git --exec-path 2>/dev/null)"
19     COMPLETION_PATH="$GIT_EXEC_PATH%/libexec/git-core"
20     COMPLETION_PATH="$COMPLETION_PATH%/lib/git-core"
21     COMPLETION_PATH="$COMPLETION_PATH/share/git/completion"
```

You will find several lines that can be altered to change the configuration of the shell prompt.

To remove the "MINGW64" you need to remove the following two lines

```
PS1="$PS1"'\[\033[35m\' # change to purple
```

```
PS1="$PS1"'$MSYSTEM ' # show MSYSTEM
```

I like to also change the first PS1 line which sets the title by removing the

```
$TITLEPREFIX:
```

and replacing it with

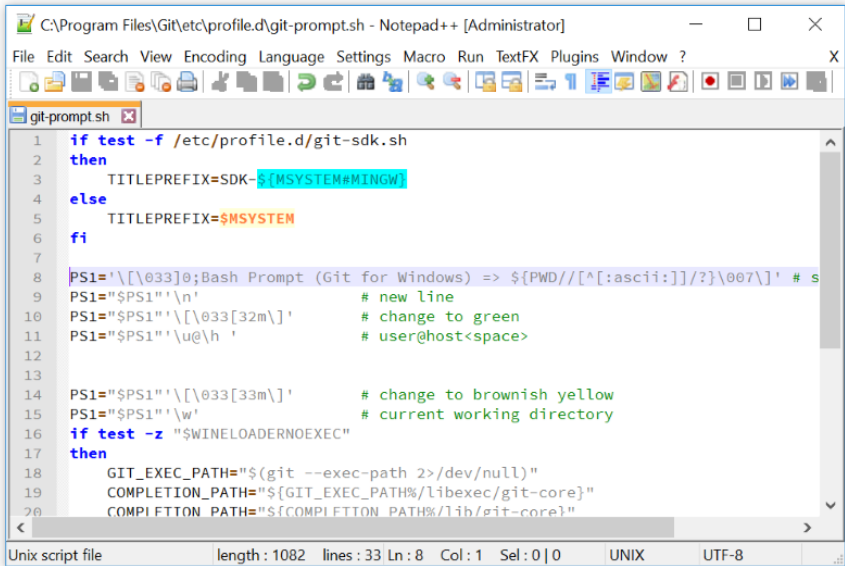
```
Bash Prompt (Git for Windows) =>
```

If you wish to set colors you will notice several places where the code looks like

```
\[033[32m\]
```

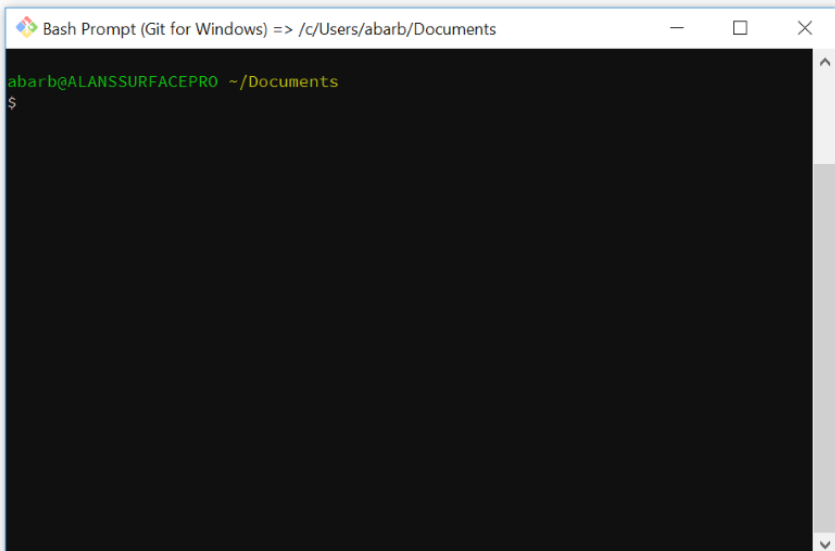
The 32 is the color code for green. Here is a table of basic color codes

Code	Color
30	Black
31	Red
32	Green
33	Yellow
34	Blue
35	Magenta
36	Cyan
37	White

A screenshot of a Notepad++ window titled "C:\Program Files\Git\etc\profile.d\git-prompts.sh - Notepad++ [Administrator]". The window shows a shell script for configuring the Git Bash prompt. The script uses conditional logic to set the prompt based on the system type (MINGW or MSYS) and includes comments for customizing the prompt's appearance with ANSI escape codes for colors and text. The status bar at the bottom indicates the file is a Unix script, 1082 bytes long, 33 lines, and is in UTF-8 encoding.

```
1 if test -f /etc/profile.d/git-sdk.sh
2 then
3     TITLEPREFIX=SDK-${MSYSTEM#MINGW}
4 else
5     TITLEPREFIX=$MSYSTEM
6 fi
7
8 PS1='\[\033]0;Bash Prompt (Git for Windows) => ${PWD//[[:ascii:]]/?}\007\' # s
9 PS1="$PS1"\n' # new line
10 PS1="$PS1"'\[\033[32m\' # change to green
11 PS1="$PS1"'\u@h \' # user@host<space>
12
13
14 PS1="$PS1"'\[\033[33m\' # change to brownish yellow
15 PS1="$PS1"'\w' # current working directory
16 if test -z "$WINELOADERNOEXEC"
17 then
18     GIT_EXEC_PATH="$(git --exec-path 2>/dev/null)"
19     COMPLETION_PATH="${GIT_EXEC_PATH%/libexec/git-core}"
20     COMPLETION_PATH="${COMPLETION_PATH%/lib/git-core}"
```

After you have saved your changes you can re-open up the bash window and you should find that your prompt now looks like this.

A screenshot of a Git Bash terminal window titled "Bash Prompt (Git for Windows) => /c/Users/abarb/Documents". The terminal shows the user's prompt as "abarb@ALANSSURFACEPRO ~ /Documents" followed by a dollar sign "\$". The background is black, and the text is green.

```
Bash Prompt (Git for Windows) => /c/Users/abarb/Documents
abarb@ALANSSURFACEPRO ~ /Documents
$
```