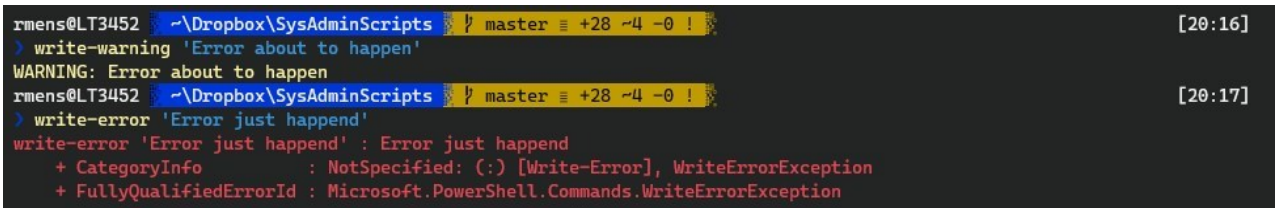
Customizing the new Windows Terminal

lazyadmin.nl/powershell/customizing-windows-terminal

January 7, 2020

Normally I start a new year with cleaning up my desktop, getting rid of all the crap that I stored on it the last 12 months or so. But this year I started with customizing my Windows Terminal.

I have written before about the new Windows Terminal and shared some tips on styling it. But in this article, we are going to take it a step further.



I noticed a PowerShell module, oh-my-posh, that is a theme engine for PowerShell. It's intended use is for ConEmu, but it works for PowerShell in general, so it can also be used with Windows Terminal.

Install Windows Terminal

If you don't have Windows Terminal already installed then you should, of course, start with installing it. Now, this is pretty easy, just go to the Windows Store and download and install it.

By using the Windows Store updates for Windows Terminal will automatically be installed.

If you have been using Windows Terminal from the beginning, you can better clear out your profiles.json. The structure for the profiles.json is changed starting from version 0.5. So make sure you make a backup of your current profile and delete everything that is it.

Installing oh-my-posh, Nerd Fonts, and POSH-Git

The next step is to install the PowerShell module oh-my-posh. This is the theming engine for PowerShell.

If you are using GIT, then you can also install POSH-Git.

Install-Module posh-git -Scope CurrentUser #OPTIONAL
Install-Module oh-my-posh -Scope CurrentUser
Run the commands above in PowerShell. This will download and install modules. You might get a warning, but you can safely click it away.

Nerd Fonts

The next step is to install the Nerd Fonts. These fonts are used for all the icons and symbols in PowerShell. Now at the moment, they are talking about adding these fonts by default in Windows Terminal, but for now, we will have to install these manually.

You can download the Nerd Fonts here from this Github repo. Make sure you download the Delugia.Nerd.Font.Complete because this also includes these fancy Powerline symbols that we are going to use later.

Customizing the Windows Terminal Profile

So we got now everything installed, now it's time to edit the Windows Terminal and PowerShell profiles. We are going to start with the PowerShell profile, to open your profile you can simply enter the following command in the terminal:

notepad $PROFILE
In your profile add the following 3 lines at the end

Import-Module posh-git #Optional - only if you are using Git
Import-Module oh-my-posh
Set-Theme paradox
This way the oh-my-posh module is loaded when the PowerShell is started and the theme is set to paradox. You can find the default theme that comes with oh-my-posh here.

The Windows Terminal Profile

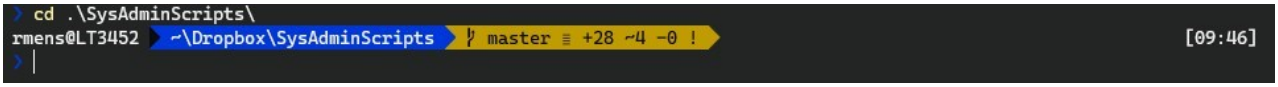
Now the default background of Windows Terminal is dark blue. That doesn't really look nice. But with Windows Terminal we can change this to whatever we like. Open the profile by pressing the following key combination `ctrl + ,`.

To change the looks of PowerShell edit the first profile with the name `Windows PowerShell` as follows:

```
"profiles":
[
{
"guid": "{61c54bbd-c2c6-5271-96e7-009a87ff44bf}",
"name": "Windows PowerShell",
"commandline": "powershell.exe",
"hidden": false,
"fontFace": "Delugia Nerd Font",
"cursorColor": "#FFFFFF",
"cursorShape": "bar",
"fontSize" : 10,
"background" : "#232524"
},
// Other profile are below here
```

Make sure you set the fontFace to `Delugia Nerd Font`, the background color to black or a dark gray `#232524`. If you want to know more about creating custom profiles you should check out this article.

Your Windows Terminal PowerShell tab should now look similar to this:



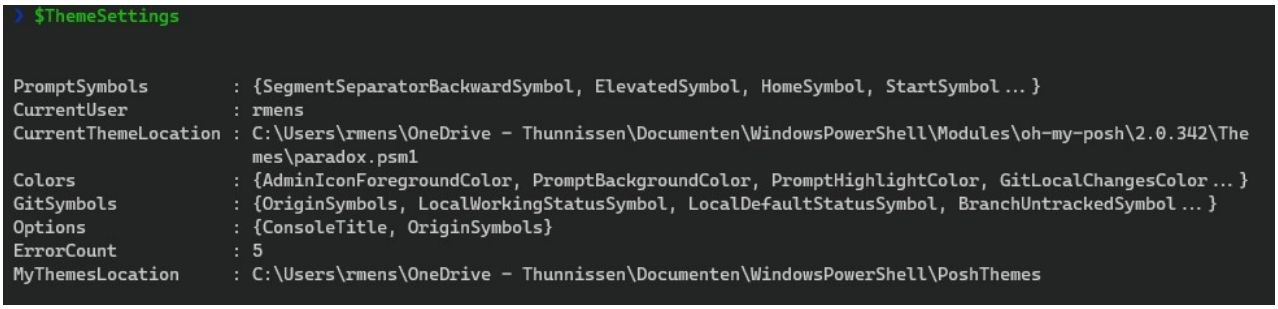
The yellow part if from Git, so if you are not using Git you won't see it 😉

Changing the segment separator

As you probably noticed in the first paragraph, I have a different separator between the path and Git. These are the PowerLine icons, but to get those we will need to create a custom theme for oh-my-posh.

Now, this is pretty simple, you can simply copy an existing one and change it. First, we will need to find the Theme files. Enter the following cmd in PowerShell:

$ThemeSettings



Here you will see the `CurrentThemeLocation` and the `MyThemesLocation`. Open the CurrentThemeLocation and copy the `paradox.psm1` file to your MyThemesLocation.

Rename the file and open it with your favorite editor. At the end of the file, you will see all the variables for the colors and symbols:

```
$sl = $global:ThemeSettings #local settings
$sl.PromptSymbols.StartSymbol = ''
$sl.PromptSymbols.PromptIndicator = [char]::ConvertFromUtf32(0x276F)
$sl.PromptSymbols.SegmentForwardSymbol = [char]::ConvertFromUtf32(0xE0C6)
$sl.Colors.PromptForegroundColor = [ConsoleColor]::White
$sl.Colors.PromptSymbolColor = [ConsoleColor]::White
$sl.Colors.PromptHighlightColor = [ConsoleColor]::DarkBlue
$sl.Colors.GitForegroundColor = [ConsoleColor]::Black
$sl.Colors.WithForegroundColor = [ConsoleColor]::DarkRed
$sl.Colors.WithBackgroundColor = [ConsoleColor]::Magenta
$sl.Colors.VirtualEnvBackgroundColor = [System.ConsoleColor]::Red
$sl.Colors.VirtualEnvForegroundColor = [System.ConsoleColor]::White
```

Here you can change the SegmentForwardSymbol to one of the PowerLines symbols. You will find the UTF code for the symbols here on the Github repo page. I have used `E0C6`.

Now to load your own theme, simply enter the following command: `Set-Theme <your-file-name>`. To load your theme when you open PowerShell you can change the PowerShell profile: `notepad $profile` and change the Set-Theme line.

If you have any suggestions on customing Windows Terminal / PowerShell further, just drop a comment below!