Bastian Fernandez

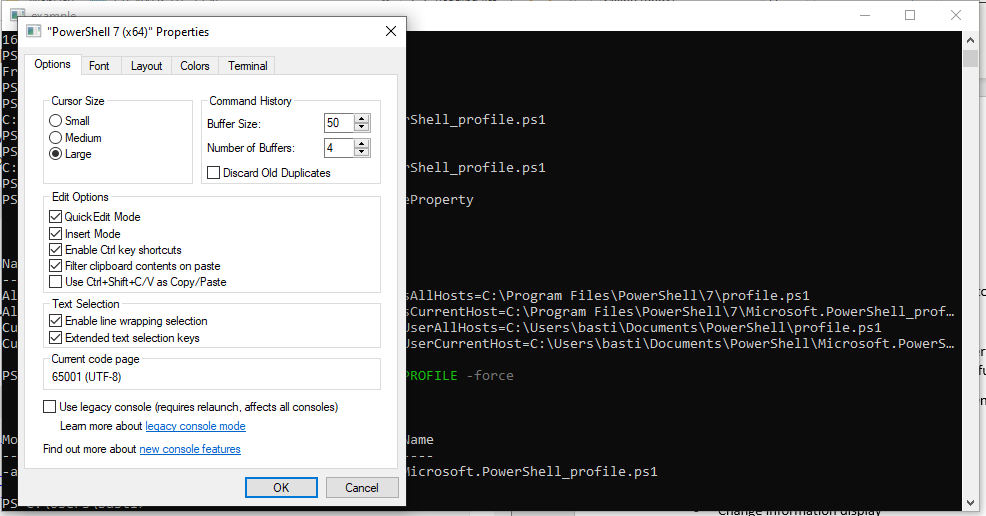
CIT361

07/15/21

Shell customization

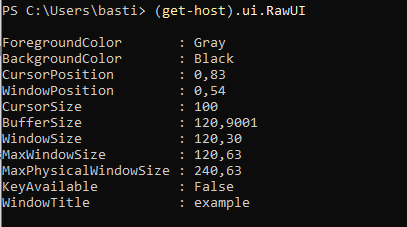
Customization means that we modify console properties so the shell can look in a different way can ether facilitate working with it or just make it more fun to use.

There are different ways to customize you consol. One of them is use the properties option of the command window. This will display many options that you can adjust like font, colors, layout with a graphical interface.



Other way to customize the console is to use PowerShell to do it. To modify console properties, we can use the follow command to access the console properties:

* **(Get-Host).UI.RawUI**

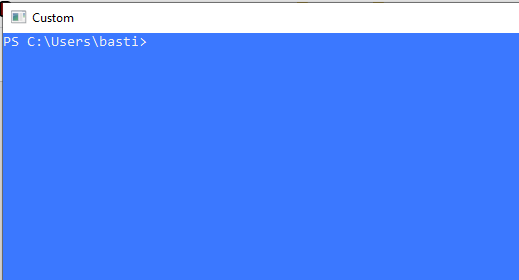


This are the current properties of the console that I have running. As you can see there are properties like ForegroundColor, BackgroundColor, CursorSize, etc. Many of these properties can be modify.

Example:

Let say that I want to change the backgroundColor and windowTItle property. We just select the property that we want to change and give them a valid value. Here is how you do it:

* **((Get-Host).UI.RawUI).BackgroundColor = “blue”**
* **((Get-Host).UI.RawUI).WindowTitle = “Custom”**



This can be very useful if you are using many sessions and you want to differentiate them, you can add a window title that describe the session or the colors that you like.

**Prompt**

Other important part that can be modify is the shell prompt. This is something that can be very useful if you want to have important information displaying in your prompt like the user, date, what folder you are in, if you are in privileged mode, use a different color, or simply you want to change the way that it looks.

To modify your prompt, you will need to change the prompt function. Here is simple example of how to change the prompt function:

* **function prompt {"This is a new prompt >>"}**



You can also modify your prompt to display the current date:

* **function prompt {"$(Get-Date)>> "}**



Now that we know how to make changes to the way that the console look like we need to learn how to make these changes be permanents and do not be erased as soon as we close the session. To accomplish that we use the PowerShell profile.

**PowerShell Profile**

With a PowerShell profile you can customize your environment and add elements to a PowerShell session. A profile is a script that runs when PowerShell starts. PowerShell support several profiles for users, but it does not create the profile for you.

To make sure that the changes that we make to our shell work every time that we open a new session, we will need to create a profile.

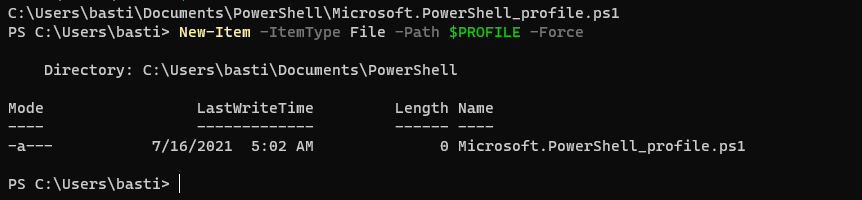
One of the things that we can custom is the prompt. To do this we need to add the prompt function to the profile of the user.

**First, I am going to show how to create the profile script.**

To find the path to the profile of the current session we use the **$PROFILE** variable. This variable stores the path of the “current user, current host” profile. As mentioned, before we know that the profile script is not created, so the next step will be to create the script.

To create the profile, we use this command:

* New-Item -ItemType File -Path <*profile-name*> -Force
* *profile-name:* here we use the $PROFILE variable



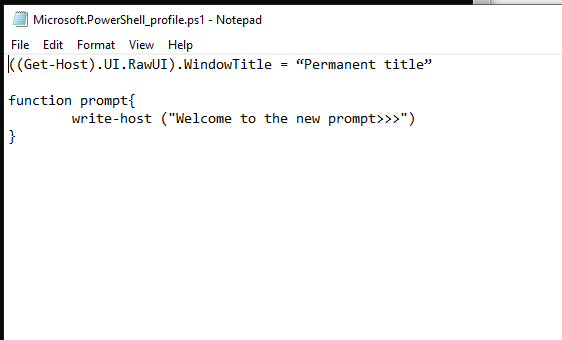
**Edit the profile to make changes in the shell**

Now that you profile is created, we can open it and add the specific changes that we want to add to the shell.

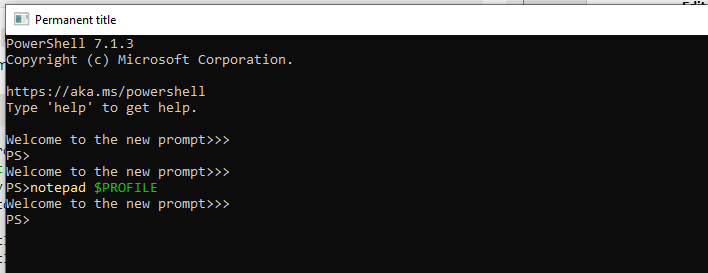
To do this you only need to go to the folder that your $PROFILE specify and edit the scrip with your favorite editor.

Example

I am going to use some of the things that we practice before. First I am going to change the window title to be “permanent title” and the prompt to say “Welcome to the new prompt”.



After having my script done and save the only thing that I need to do is close the console and open a new one. The new console will display the new properties.



**Hands On.**

For the exercise you will custom your own console.

1. Create a profile so you can add your command and functions to change your console
   * New-Item -ItemType File -Path <*profile-name*> -Force
2. Using your favorite editor, change the window title to something of your election.
   * ((Get-Host).UI.RawUI).WindowTitle = “example”
3. Now create the function prompt. Make the prompt be:
   1. Your name
   2. Computer Name
   3. Date of the week
   4. Separate the parameters with a ‘:’
   5. Add “>>” at the end with a space “ “

Hint:

* + Use the write-host to format the output of the prompt
    - Ex:



* + Remember to close your console and open a new one to see the changes.

1. Take a screenshot and paste it in here

Text

Description automatically generated

Finally delete the profile file that you create to get rid of the modification that you did to your console.