# [vscode-PowerShell] Thing: Bug Report, Star Date: [10/22/2022 10:07:54]

I've been having issues getting the PowerShell Extension to load the regular desktop version of PowerShell upon loading the extension/vscode. I've researched the problem to a minor degree, but none of the suggestions I have found, have worked. Here's a copy+paste of my settings.json file...

## settings.json

```json
{
    "workbench.colorTheme": "BossMode",
    "terminal.integrated.defaultProfile.windows": "PowerShell (v5)",
    "terminal.integrated.profiles.windows": {
        "Command Prompt": {
            "path": [
                "${env:windir}\\Sysnative\\cmd.exe",
                "${env:windir}\\System32\\cmd.exe"
            ],
            "args": [],
            "icon": "terminal-cmd"
        },
        "Git Bash": {
            "source": "Git Bash"
        },
        "PowerShell (v5)": {
            "path": "${env:windir}\\system32\\WindowsPowerShell\\v1.0\\powershell.exe"
        },
        "PowerShell (v6+)": {
            "path": "C:\\Program Files\\PowerShell\\7\\pwsh.exe"
        },
    },
    "workbench.editor.untitled.hint": "hidden",
    "window.zoomLevel": -1
}
```

settings.json

## $PSVersionTable

Here's a copy paste of the trusty ol' $PSVersionTable output. As you can see, there appears to be some confusion under the hood. The PS Int'd Console in VSCode is loading “Core” rather than “Desktop”.

```
PS Prompt:\> $PSVersionTable

Name                           Value
----                           -----
PSVersion                      7.2.7
PSEdition                      Core
GitCommitId                    7.2.7
OS                             Microsoft Windows 10.0.19044
Platform                       Win32NT
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0…}
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
WSManStackVersion              3.0

PS Prompt:\>
```

$PSVersionTable

```
  Error/PSReadLine /------------------------------------------------------------------------------------------\
/------------------
```

The reason I would prefer to use the “Desktop” version of PowerShell for the Integrated Console, is because I keep experiencing an error when I run some of my code via pwsh.exe. It'll say something like:

```
Oops, something went wrong.  Please report this bug with the details below.
Report on GitHub: https://github.com/lzybkr/PSReadLine/issues/new
-----------------------------------------------------------------------
Last 200 Keys:
 l l . P S C o n s o l e R e a d L i n e . R e a d L i n e ( R u n s p a c e Space r u n s p a c e ,
 Space Enter Space Space Space E n g i n e I n t r i n s i c s Space e n g i n e I n t r i n s i c s )
 Enter
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - Enter
 ' @ Space | Space W r i t e - C o m m e n t Space - I Space 4 Space | Space S e t - C l i p b o a r d
 Enter

Exception:
System.ArgumentOutOfRangeException: The value must be greater than or equal to zero and less than the
console's buffer size in that dimension.
Parameter name: top
Actual value was -17.
   at System.Console.SetCursorPosition(Int32 left, Int32 top)
   at Microsoft.PowerShell.PSConsoleReadLine.ReallyRender(RenderData renderData, String defaultColor)
   at Microsoft.PowerShell.PSConsoleReadLine.ForceRender()
   at Microsoft.PowerShell.PSConsoleReadLine.Insert(Char c)
   at Microsoft.PowerShell.PSConsoleReadLine.SelfInsert(Nullable`1 key, Object arg)
   at Microsoft.PowerShell.PSConsoleReadLine.ProcessOneKey(ConsoleKeyInfo key,
      Dictionary`2 dispatchTable, Boolean ignoreIfNoAction, Object arg)
   at Microsoft.PowerShell.PSConsoleReadLine.InputLoop()
   at Microsoft.PowerShell.PSConsoleReadLine.ReadLine(Runspace runspace,
      EngineIntrinsics engineIntrinsics)
-----------------------------------------------------------------------
```

I've formatted the code slightly differently than it comes out, because it exceeded the design width of this particular document, and that's no bueno. The thing is, this error occurs in Windows Terminal as well.

Maybe I need to submit the bug report at the listed at: | https://github.com/lzybkr/PSReadLine/issues/new |

But- the problem comes down to this... The VSCode PowerShell Integrated Console is NOT producing those errors, at all. It's just loading the wrong PS version. When I highlight the code, press F8, and it processes it...?

It's NEARLY instant.
When it is NOT the PowerShell Integrated Console, then when I highlight the code I want to run, and hit F8, then what happens is that it treats all of the input as MULTIPLE SPLIT STRINGS, rather than a SINGLE HERESTRING.

So, allow me try to showcase HOW the console is processing the input...

```
                                                                                           ________________/
\_______________________________________________________________________________________/ $PSVersionTable
  Example /-----------------------------------------------------------------------------------------------\
/---------
```

```
$Content = @(0..15 | % { "Line {0:d2}" -f $_ })
$Content
$Content.GetType().Name
```

```
PS Prompt:\> $Content
Line 00
Line 01
Line 02
Line 03
Line 04
Line 05
Line 06
Line 07
Line 08
Line 09
Line 10
Line 11
Line 12
Line 13
Line 14
Line 15
PS Prompt:\> $Content.GetType().Name
Object[]
PS Prompt:\>
```

As you can see, the OUTPUT looks like all of those lines are an (Object[]/Object array).
That's how the PSReadline thing in the non-(PowerShell Integrated Console) is rendering the output/script,
whereby causing the error messages way up above.

```
$Content = @(0..15 | % { "Line {0:d2}" -f $_ }) -join "`n"
$Content
$Content.GetType().Name
```

```
PS Prompt:\> $Content
Line 00
Line 01
Line 02
Line 03
Line 04
Line 05
Line 06
Line 07
Line 08
Line 09
Line 10
Line 11
Line 12
Line 13
Line 14
Line 15
PS Prompt:\> $Content.GetType().Name
String
PS Prompt:\>
```

This is how the PowerShell Integrated Console actually processes the content in the script editor, in nearly an
identical way, to how the PowerShell ISE processes the content of its' script pane.

The end result, is that highlighting and executing some code in a different console takes a lot longer.
If the script is like, 100 lines...? It might take the same amount of time as the above example.
If the script is like, 60,000 lines...? It may take at least (1) full 60-minute hour, and no less than that.
If the script is like, 1,000,000 lines...? Then you'll be waiting (1) full, entire, Earth-cycle (week/month).

Now, I'm hyperbolizing that to be cheeky, but- allow me to explain what a (week/month) is, in the next section...

```
                                                                                                  __________________/
\____________________________________________________________________________________________________/ Error/PSReadLine
  Week/Month /¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯\
/¯¯¯¯¯¯¯¯¯¯¯¯
```

In the age of chaos, (2) factions battled for dominance.
On one side...? You had castles made of brick, towering over the shores of Lorderon. Humans running with haste,
running to man the cannons on the top of the castle walls, awaiting the Tides of Darkness...

On the other side...? Your standard-issue Tides of Darkness, and a really really Dark Portal.
Ships full of ogres, grunts, axethrowing trolls with mohawks, and goblins that perform suicide attacks.

None of that has anything to do with this bug report, except... I'm about to talk about the term week/month.

You see, back in the early days of the internet, people would say stuff like “Man, Warcraft II is an awesome
game. It's just too bad that the game is really laggy on a dial-up modem.”

Today, Blizzard/Activision has the trusty ol' Battle.net. But back then...? That didn't exist until Starcraft
was released in 1998. Anyway, sometimes when someone would play this game I've just described, WAY back in like,

1996...? People would get wicked pissed if they lost because of how laggy the game was. Because people would spend hours and hours playing one another, and if they spent an entire hour building a base, mining gold, chopping down trees, listening to the peon's saying “Zug Zug”, and “Daboo”...?

WELL... everybody would rightfully get pissed off at how many week/months they invested into this cool game, only for the enemy player to have had a strategy that didn't need to consider the lagginess of the game.

The unholy term (week/month), is a shorthand terminology for the amount of time between (1) week - (1) month.

The guys who worked at Blizzard Entertainment knew that many children AND adults, were out there... playing these high stakes games against their friends over phone lines, sweat beading from their brows, pouring their hearts and souls into accumulating resources, and laying total waste to their opponents.

The guys at Blizzard Entertainment didn't like the idea of their game, taking entire (week/month)'s out of their loyal fans lives, to play this game over the internet against their (friends/buddies), only to consistently lose. Because... when people consistently lose at a game that is FUN to play, BUT- they can't win at the game because of the lagginess...? it's gonna cause their beloved fans to “abandon ship”, and Blizzard Entertainment didn't really like that idea in the least.

Nah. So, that's when at some point in 1998...? They came up with the idea for Battle.net.
Battle.net actually offloaded some of the lagginess of the game, as it allowed the server to exist on the internet, rather than for the modems to send/receive additional instructions and then run code on each end, to reproduce the game state. In the early days of the internet, many programmers went head-to-head, trying to build the BEST game networking code in existence... Many have tried, many have failed... Only a few succeeded.

The idea of Battle.net succeeded. Because... once they came up with the idea of Battle.net...?
They introduced the idea of using the INTERNET to offload some of the processing overhead AND the network stats. Starcraft was the first game that truly allowed people to CONTINUE using their MODEMS to DIRECT CONNECT with one another, featuring IPX/SPX, IPV4... OR, <check out Battle.net, dudeface>. Even Quake II/Quake III Arena, which had the BEST networking programming known to mankind for it's time...? It did not have the option of joining a centralized server.

Once Battle.net was a thing...?
People realized that the unholy term (week/month), could be ACCURATELY translated into them just being bad at the game whenever they lost. And, that's important. Sometimes people will do everything humanly possible to AVOID accepting that they just... sucked at a particular game, and then the term “rage quit” was born.

```
                                                                                                    ____________/
\__________________________________________________________________________________________________/ Week/Month
  Conclusion /¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯\
/¯¯¯¯¯¯¯¯¯¯¯¯
```

Now, how the hell does any of this have to do with this bug report...?
Uh- whenever I highlight the code in Visual Studio Code, I sometimes have to wait entire (week/month)'s, for the code to be processed into the engine, and if there was a single error in that entire code...? Oh no. Gotta do it all over again.

So, in reality, then I have to consider building the script to save itself to a FILE, and then just import the module THAT way, so that it can avoid wasting entire (week/month)'s of MY particular life...

But, it doesn't change the fact that I still really enjoy using Visual Studio Code.

```
                                                                                                    ____________/
\__________________________________________________________________________________________________/ Conclusion
```

```
 ________________________________________________________
|¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯|
|                                     Michael C. Cook Sr. |
|                                       Security Engineer |
|                                    Secure Digits Plus LLC |
|________________________________________________________|
 ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯
```