

Take Your PowerShell Profile on the Road

 mattmcnabb.github.io/portable-profile

As my PowerShell skills have progressed I began to use profile scripts to customize and extend the PowerShell experience. I tend to work from a few different computers at work, at home, and sometimes on the road and keeping these profiles in sync was often a hassle. Eventually I decided that what I needed was a way to automate the syncing of my profile on any machine that I use regularly.

PowerShell Profiles

A PowerShell profile is nothing more than a script that is run by default any time you load a PowerShell host. These scripts can contain any PowerShell code you would like to have dot-sourced into your session. There are six different profiles that will be run depending on the host and third-party hosts can add their own profile as well. I typically use the 'Current User, All Hosts' profile which ensures that whether I open the console or the ISE I will still have my customized experience.

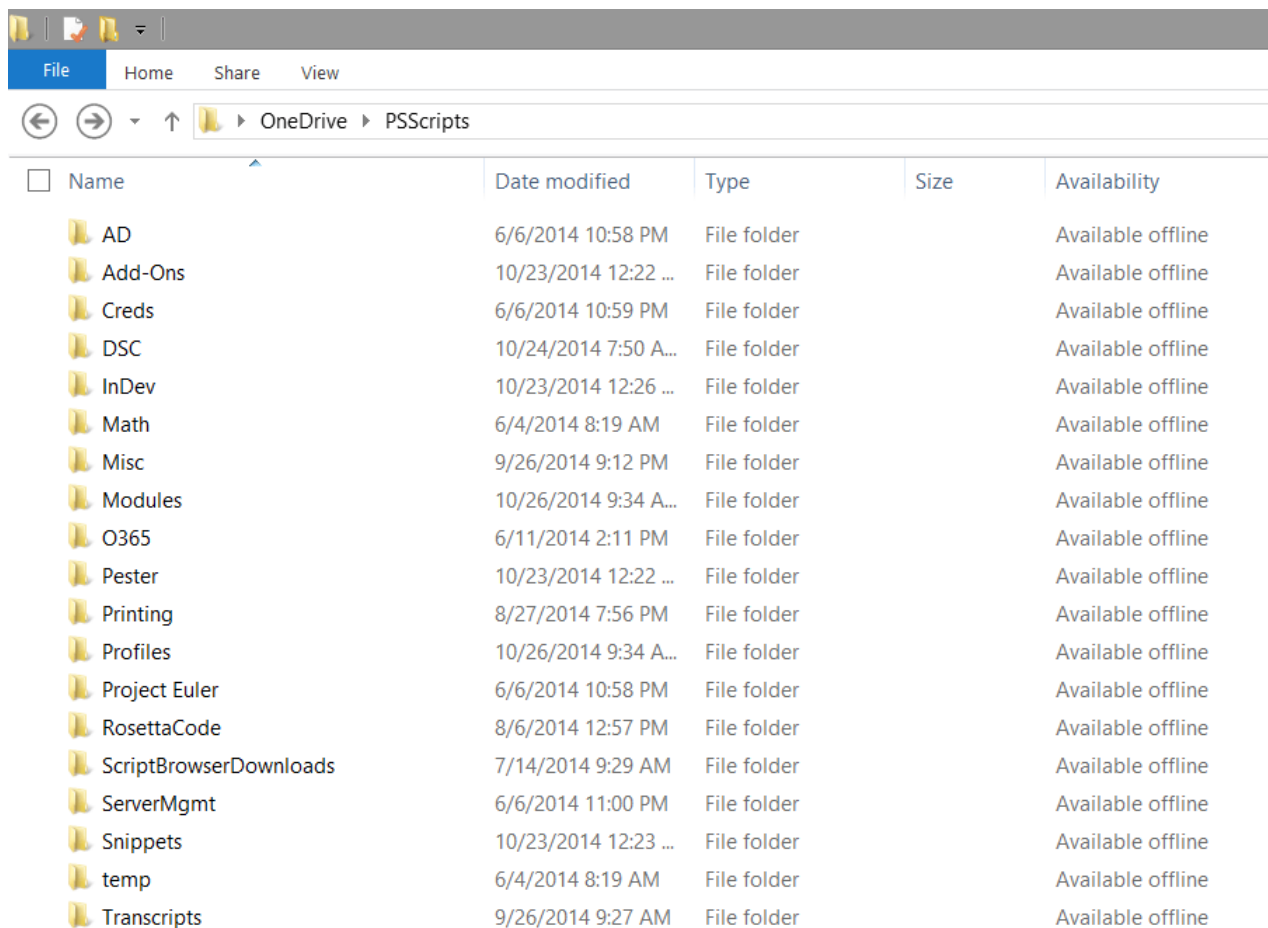
Making Your Profile Portable

To make my profile more manageable across different devices I took two approaches:

1. I broke up my profile script into several files which makes it easier to customize the console host or the ISE but still keep all the files in the same location.
2. I leveraged Microsoft OneDrive to manage the syncing of the profile scripts. With this I can make changes to my profile in one place and the changes will go with me.

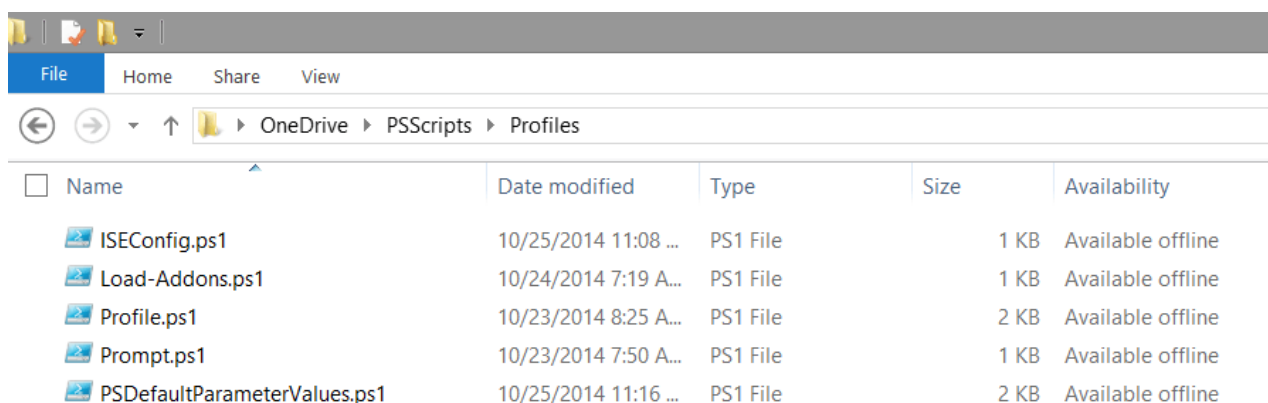
OneDrive as a Script Library

I had already been using OneDrive as a way to store scripts and access them from anywhere, so it was a natural progression to begin storing my profile scripts there as well. I keep an organized script library in a OneDrive folder that looks something like this:



<input type="checkbox"/> Name	Date modified	Type	Size	Availability
AD	6/6/2014 10:58 PM	File folder		Available offline
Add-Ons	10/23/2014 12:22 ...	File folder		Available offline
Creds	6/6/2014 10:59 PM	File folder		Available offline
DSC	10/24/2014 7:50 A...	File folder		Available offline
InDev	10/23/2014 12:26 ...	File folder		Available offline
Math	6/4/2014 8:19 AM	File folder		Available offline
Misc	9/26/2014 9:12 PM	File folder		Available offline
Modules	10/26/2014 9:34 A...	File folder		Available offline
O365	6/11/2014 2:11 PM	File folder		Available offline
Pester	10/23/2014 12:22 ...	File folder		Available offline
Printing	8/27/2014 7:56 PM	File folder		Available offline
Profiles	10/26/2014 9:34 A...	File folder		Available offline
Project Euler	6/6/2014 10:58 PM	File folder		Available offline
RosettaCode	8/6/2014 12:57 PM	File folder		Available offline
ScriptBrowserDownloads	7/14/2014 9:29 AM	File folder		Available offline
ServerMgmt	6/6/2014 11:00 PM	File folder		Available offline
Snippets	10/23/2014 12:23 ...	File folder		Available offline
temp	6/4/2014 8:19 AM	File folder		Available offline
Transcripts	9/26/2014 9:27 AM	File folder		Available offline

I keep scripts in several folders based on topic and have a folder just for modules, one for snippets and the one we are most interested in now is the one for profiles. Inside there are multiple scripts that are launched any time I open a PowerShell console or the ISE:



<input type="checkbox"/> Name	Date modified	Type	Size	Availability
ISEConfig.ps1	10/25/2014 11:08 ...	PS1 File	1 KB	Available offline
Load-Addons.ps1	10/24/2014 7:19 A...	PS1 File	1 KB	Available offline
Profile.ps1	10/23/2014 8:25 A...	PS1 File	2 KB	Available offline
Prompt.ps1	10/23/2014 7:50 A...	PS1 File	1 KB	Available offline
PSDefaultParameterValues.ps1	10/25/2014 11:16 ...	PS1 File	2 KB	Available offline

How It Works

So to make this work for you there are only two things you need to do on each computer that you regularly work on:

1. Configure OneDrive - If you are on Windows 7 this means downloading and installing OneDrive. On Windows 8.1 you just need to sign in and set up a folder to store your profile scripts.

2. Create a simple PowerShell profile in one of the default locations. The one I use is located in my Documents folder at `.\WindowsPowerShell\profile.ps1`. Here is all the code I use in that file:

```
$DirScripts = "$env:USERPROFILE\OneDrive\PSscripts"
```

```
. "$DirScripts\profiles\Profile.ps1"
```

[view raw 1.ps1](#) hosted with ❤ by [GitHub](#)

This creates a variable with the path to my OneDrive scripts library and then 'dot-sources' my primary profile script from that library. The first line is not absolutely necessary, but I like to have my scripts library easily accessible from the command line and keeping this in a variable does that.

When the profile script runs it calls the scripts above in the profiles folder and runs a number of commands that create variables, modify my module path, and load modules and add-ons. With this method I can update my profile scripts in one location and the changes go with me wherever I go. Pretty sweet, huh?

File Syncing

OneDrive in Windows 8.1 uses selective file syncing which means that files are only downloaded on demand - if you haven't accessed a file then you just get a placeholder. PowerShell can't really deal with this so if you try to launch a profile script it will fail the first time since the file has to download before it can be run. I get around this by manually marking my script library folder to always be available offline.

Do I Have to Use OneDrive?

Probably not. I have used OneDrive since it was called 'Live Mesh' and love the service. It offers quite a bit of space for free and having it built right in to Windows and Windows Phone is a nice feature. Sugarsync and Dropbox are really nice too and easy to manage so I am certain you could use these as well.

In my next post I'll go over what is actually in my profile scripts. Thanks for reading!