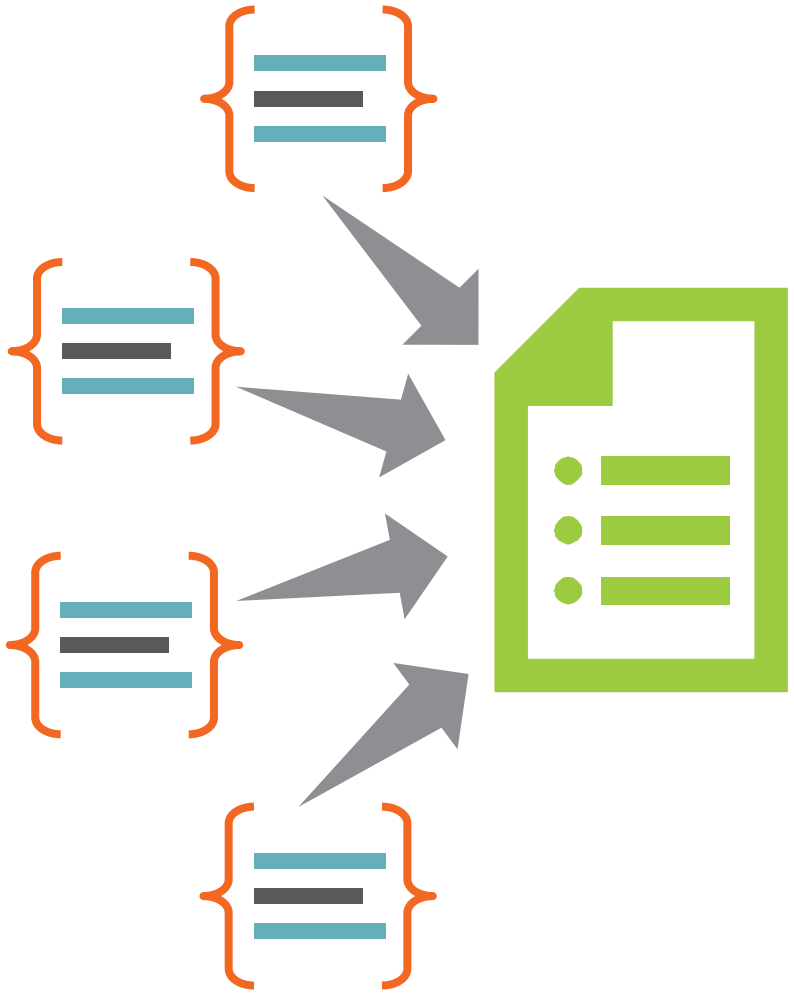
Use Modules to Create a Tool Belt

Adam Bertram

adamtheautomator.com | Twitter: @adbertram

pluralsight

What's a Module?



- A Module is just a grouping of like functions all contained in a single file
- Typically created around a central application
- Popular modules are ActiveDirectory, DnsServer, SQL, etc
- Created to easily share code with others
- Most of the modules provided by vendors do not need installation and configuration

pluralsight

Types of Modules

Script

- PSM1 file extension
- Contains various functions and sometimes variables.
- Simplest and most often used

Binary

- Created by software developers.
- Compiled code presented as a DLL file

Manifest

- Contains an associated manifest
- PSD1 file extension
- Contains version, prerequisites and/or various notes about the module

Dynamic

- Never exist as a file at all
- Created/used on-the-fly in code
- Always stored in RAM
- Perform ad-hoc functions inside scripts

pluralsight

Creating a Module

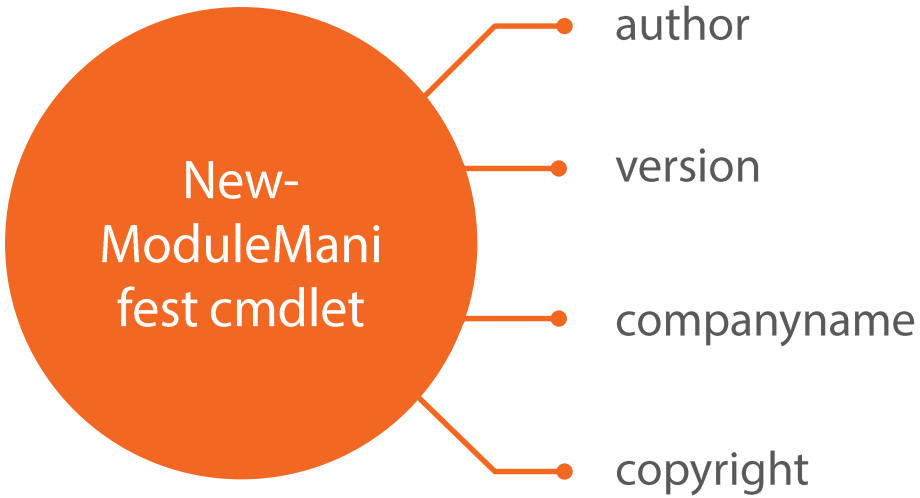
1. Create a text file with a PSM1 extension to create a simple script module

2. Copy and paste existing scripts into your module and wrap function brackets

3. You can also have variables inside your module to be used by the module itself or by the user

pluralsight

Creating a Manifest



Contains a specific structure that includes information about the module.

You can't just open up notepad and save the file with a PSD1 extension

Contains a hashtable that contains various variables such as the name of the variable, the author, minimum Powershell version

Manifest should be placed in the same folder as your PSM1 file

pluralsight

Module Location

Microsoft recommended user based location

C:\Users\$Username\Documents\WindowsPowerShell\Modules

System-level path reserved for system-level modules

C:\Windows\System32\WindowsPowerShell\v1.0\Modules

$env:PsModuleInfo

- Environment variable to find modules
- Contains one or more user defined paths

pluralsight

Using Your Module

Importing it with Import-Module or by using PowerShell's auto-loading feature

Run any of the functions available in the module or use any of the variables inside the module

pluralsight

Takeaways

Build tools and modules and not simple scripts

Modules make functions readily available and shareable

pluralsight

Summary

Creating a module

Creating a manifest in the same location

Importing and using modules

Have your tool at your fingertips!

pluralsight