# GET MORE FROM YOUR TOOLING

TURNING COMMAND-LINE TOOLS INTO CMDLETS USING CRESCENDO

## PRESENTERS

Jim Truher

Sr. Software Engineer

An original PowerShell developer

Sean Wheeler

Sr. Content Developer

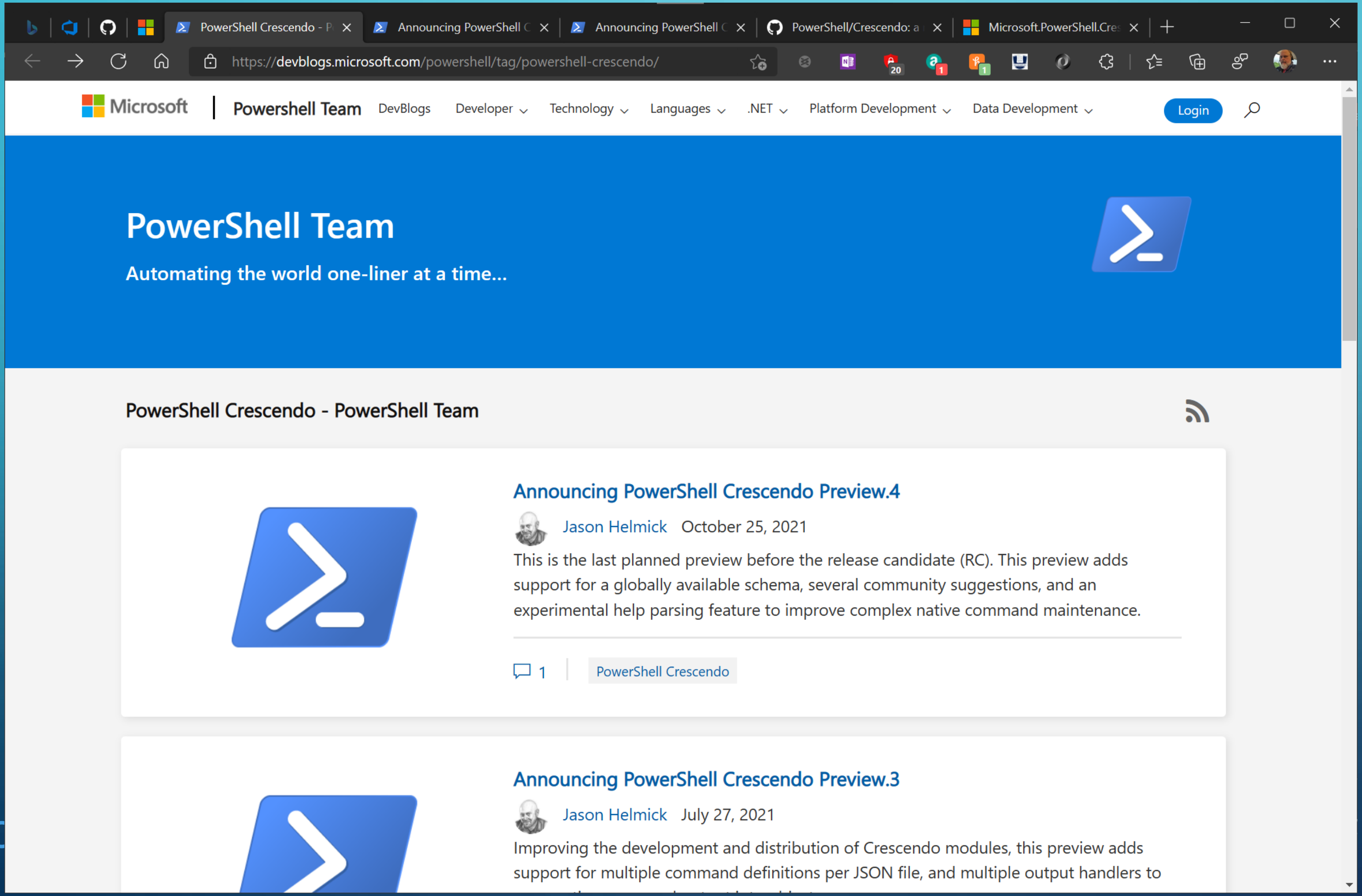
Lead Writer for PowerShell

## MY 5 STEPS TO CREATING A MODULE

1. Getting started reading the blogs
2. Choose the native command - `VSSAdmin.exe` on Windows
3. Create the output parsers
4. Create the cmdlet configuration file
5. Export the configuration to a module

DEMO TIME!

# STEP 1 – READ THE BLOGS!



PowerShell Team

Automating the world one-liner at a time…

PowerShell Crescendo - PowerShell Team

**Announcing PowerShell Crescendo Preview.4**

Jason Helmick October 25, 2021

This is the last planned preview before the release candidate (RC). This preview adds support for a globally available schema, several community suggestions, and an experimental help parsing feature to improve complex native command maintenance.

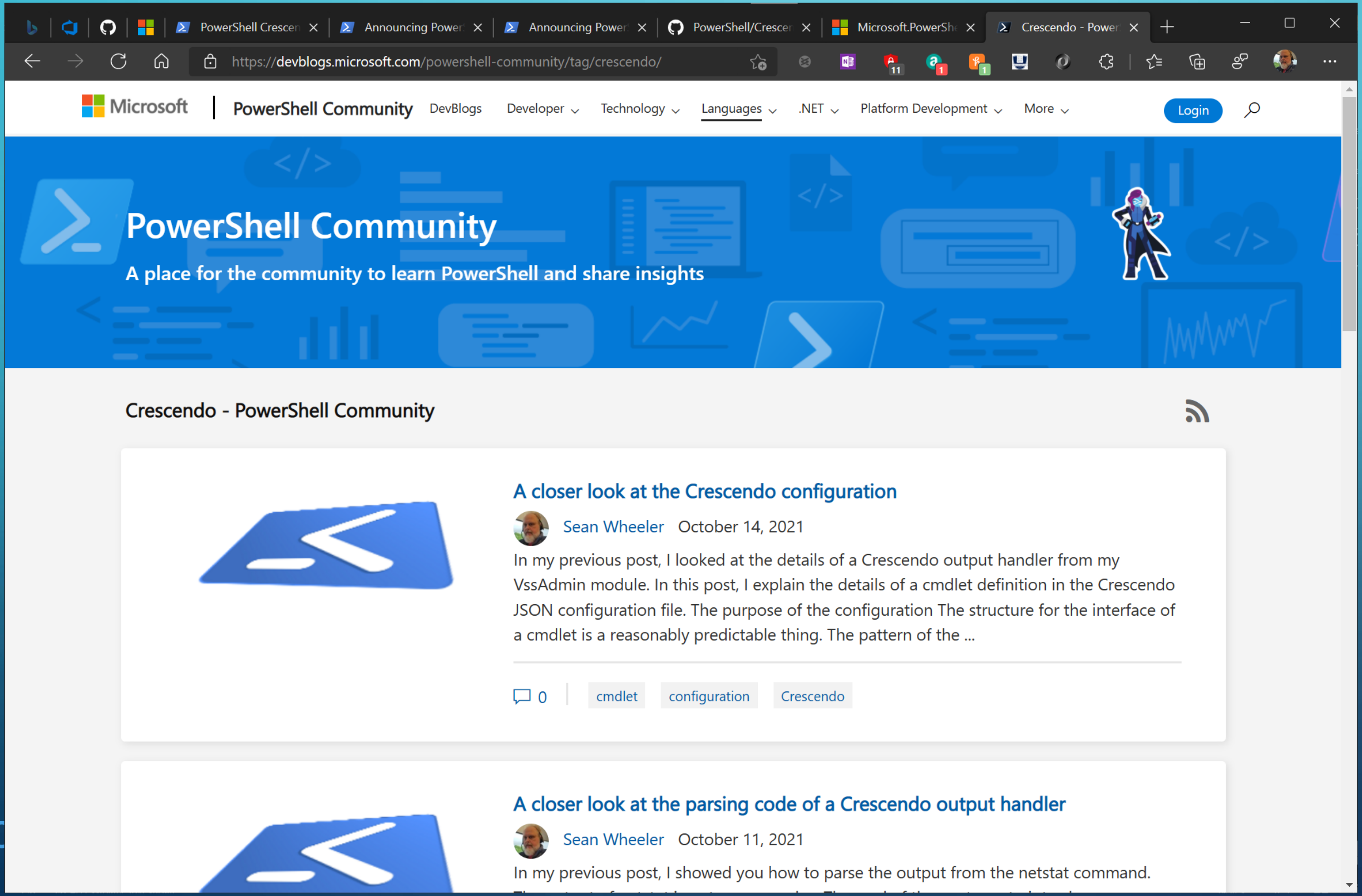
1 | PowerShell Crescendo

**Announcing PowerShell Crescendo Preview.3**

Jason Helmick July 27, 2021

Improving the development and distribution of Crescendo modules, this preview adds support for multiple command definitions per JSON file, and multiple output handlers to

# STEP 1 – READ THE BLOGS!



Crescendo - PowerShell Community

**A closer look at the Crescendo configuration**

Sean Wheeler October 14, 2021

In my previous post, I looked at the details of a Crescendo output handler from my VssAdmin module. In this post, I explain the details of a cmdlet definition in the Crescendo JSON configuration file. The purpose of the configuration The structure for the interface of a cmdlet is a reasonably predictable thing. The pattern of the ...

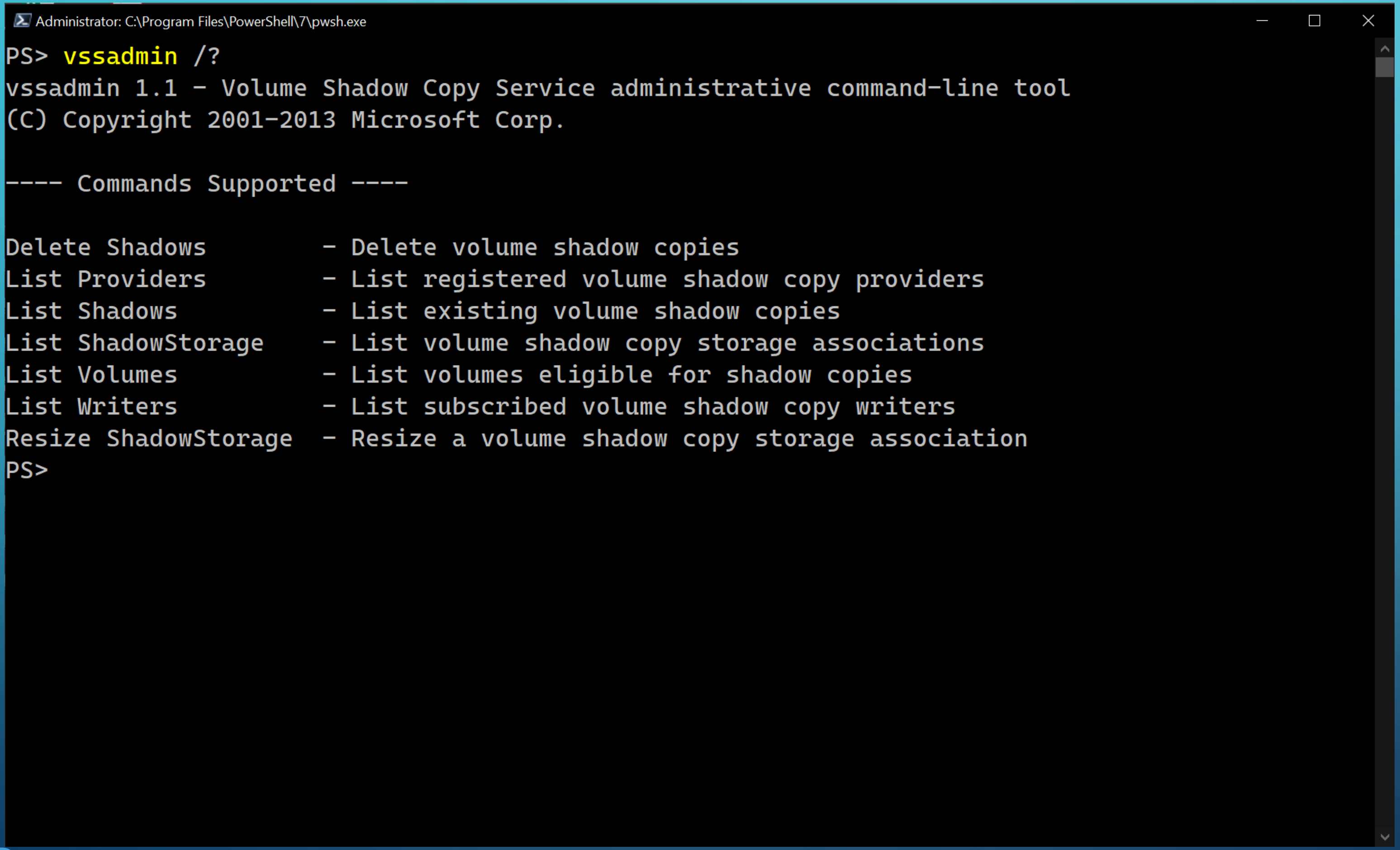
0 cmdlet configuration Crescendo

**A closer look at the parsing code of a Crescendo output handler**

Sean Wheeler October 11, 2021

In my previous post, I showed you how to parse the output from the netstat command.

## STEP 2 - CHOOSE THE NATIVE COMMAND



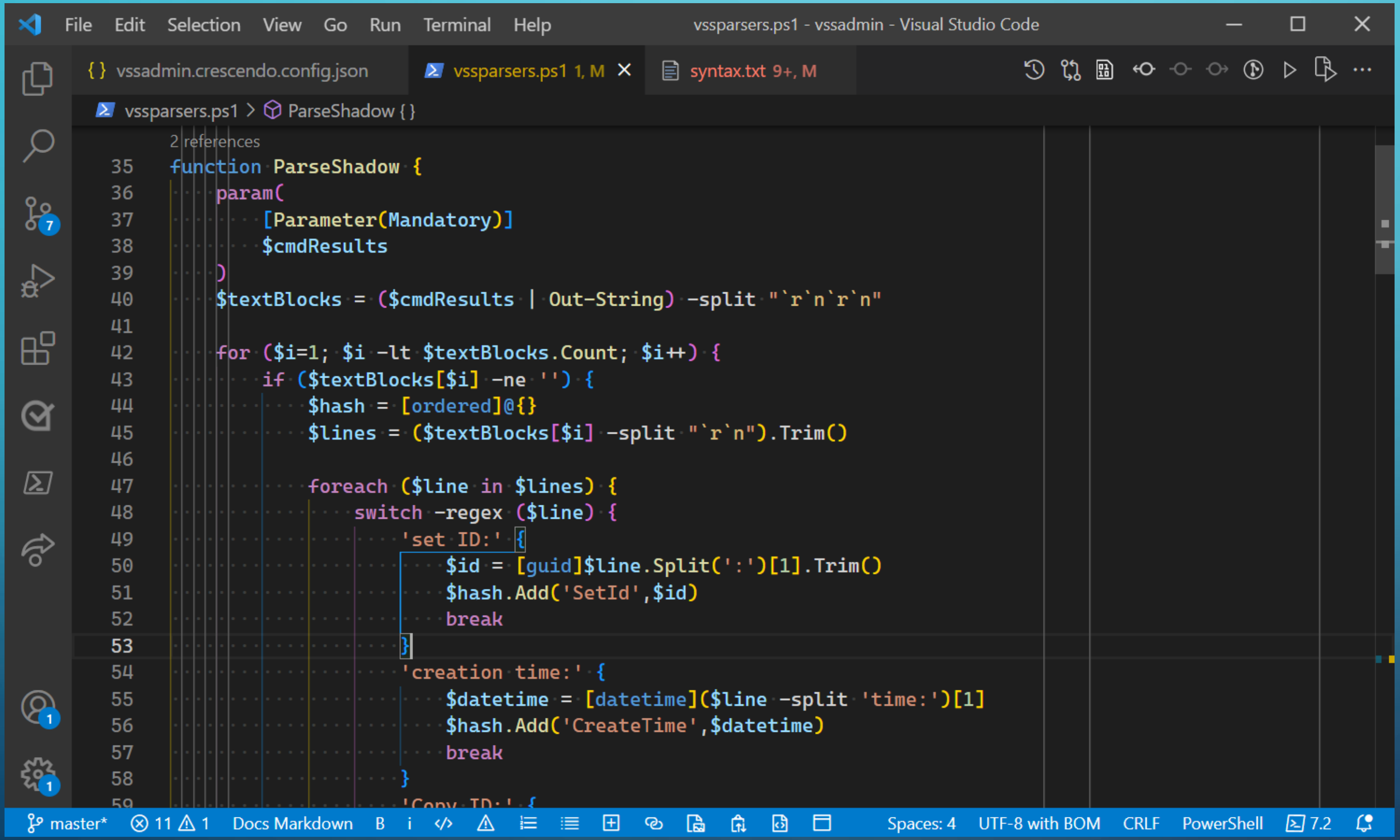
```
PS> vssadmin /?
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.

---- Commands Supported ----

Delete Shadows        - Delete volume shadow copies
List Providers        - List registered volume shadow copy providers
List Shadows          - List existing volume shadow copies
List ShadowStorage    - List volume shadow copy storage associations
List Volumes          - List volumes eligible for shadow copies
List Writers          - List subscribed volume shadow copy writers
Resize ShadowStorage  - Resize a volume shadow copy storage association
PS>
```

- Start with the help
- Collect output samples

# STEP 3 - CREATE THE OUTPUT PARSERS


File Edit Selection View Go Run Terminal Help
vssparsers.ps1 - vssadmin - Visual Studio Code
vssadmin.crescendo.config.json
vssparsers.ps1 1, M
syntax.txt 9+, M
vssparsers.ps1 > ParseShadow { }
2 references
master* ⊗ 11 ⚠ 1 Docs Markdown Spaces: 4 UTF-8 with BOM CRLF PowerShell 7.2

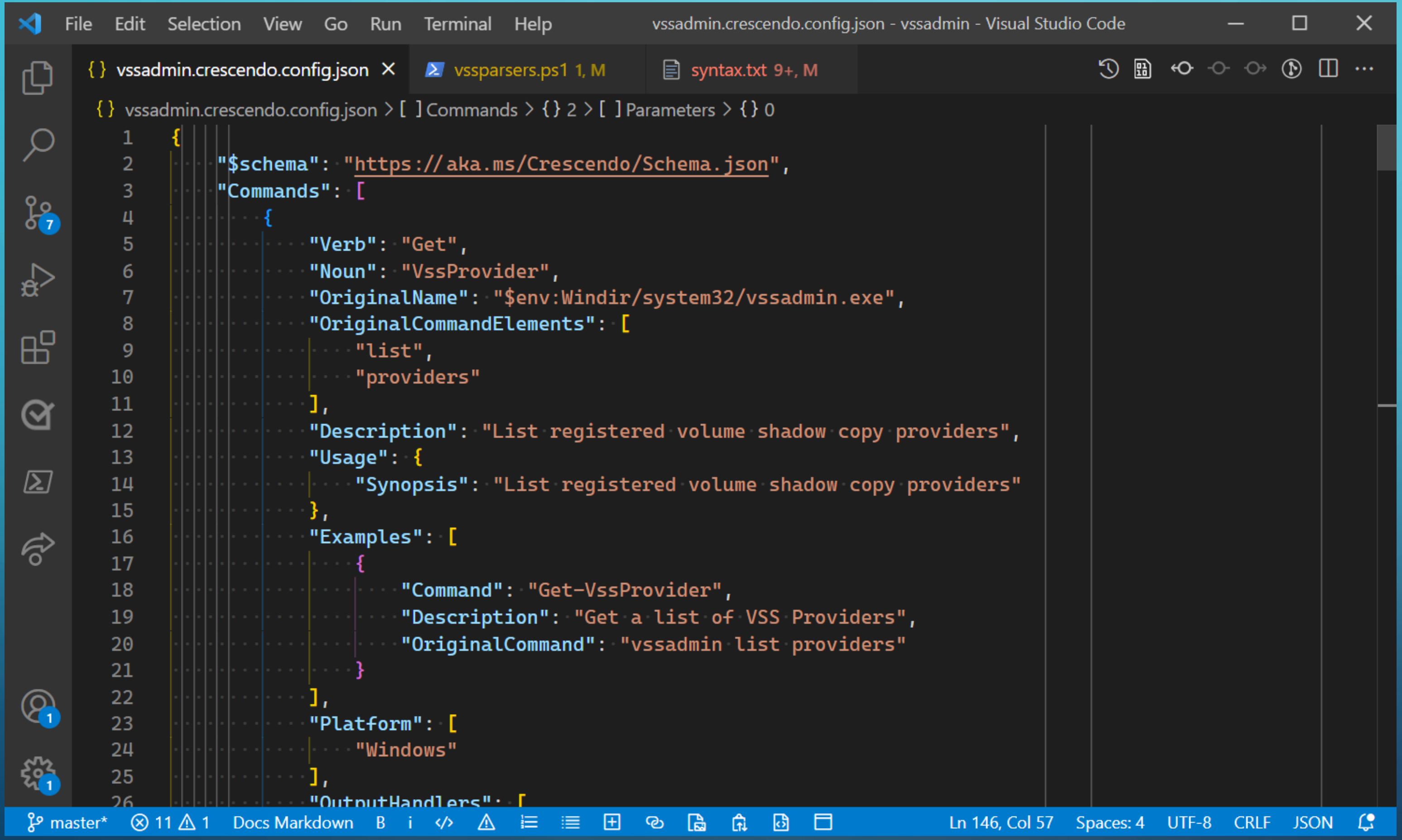
```
function ParseShadow {
    param(
        [Parameter(Mandatory)]
        $cmdResults
    )
    $textBlocks = ($cmdResults | Out-String) -split "`r`n`r`n"

    for ($i=1; $i -lt $textBlocks.Count; $i++) {
        if ($textBlocks[$i] -ne '') {
            $hash = [ordered]@{}
            $lines = ($textBlocks[$i] -split "`r`n").Trim()

            foreach ($line in $lines) {
                switch -regex ($line) {
                    'set ID:' {
                        $id = [guid]$line.Split(':')[1].Trim()
                        $hash.Add('SetId',$id)
                        break
                    }
                    'creation time:' {
                        $datetime = [datetime]($line -split 'time:')[1]
                        $hash.Add('CreateTime',$datetime)
                        break
                    }
                    'Copy ID:' {
```

- This is the hard part
- Lots of tools available - Split(), Trim(), Replace(), regular expressions (with -match and switch), etc.
- Parsers must be discoverable by Get-Command

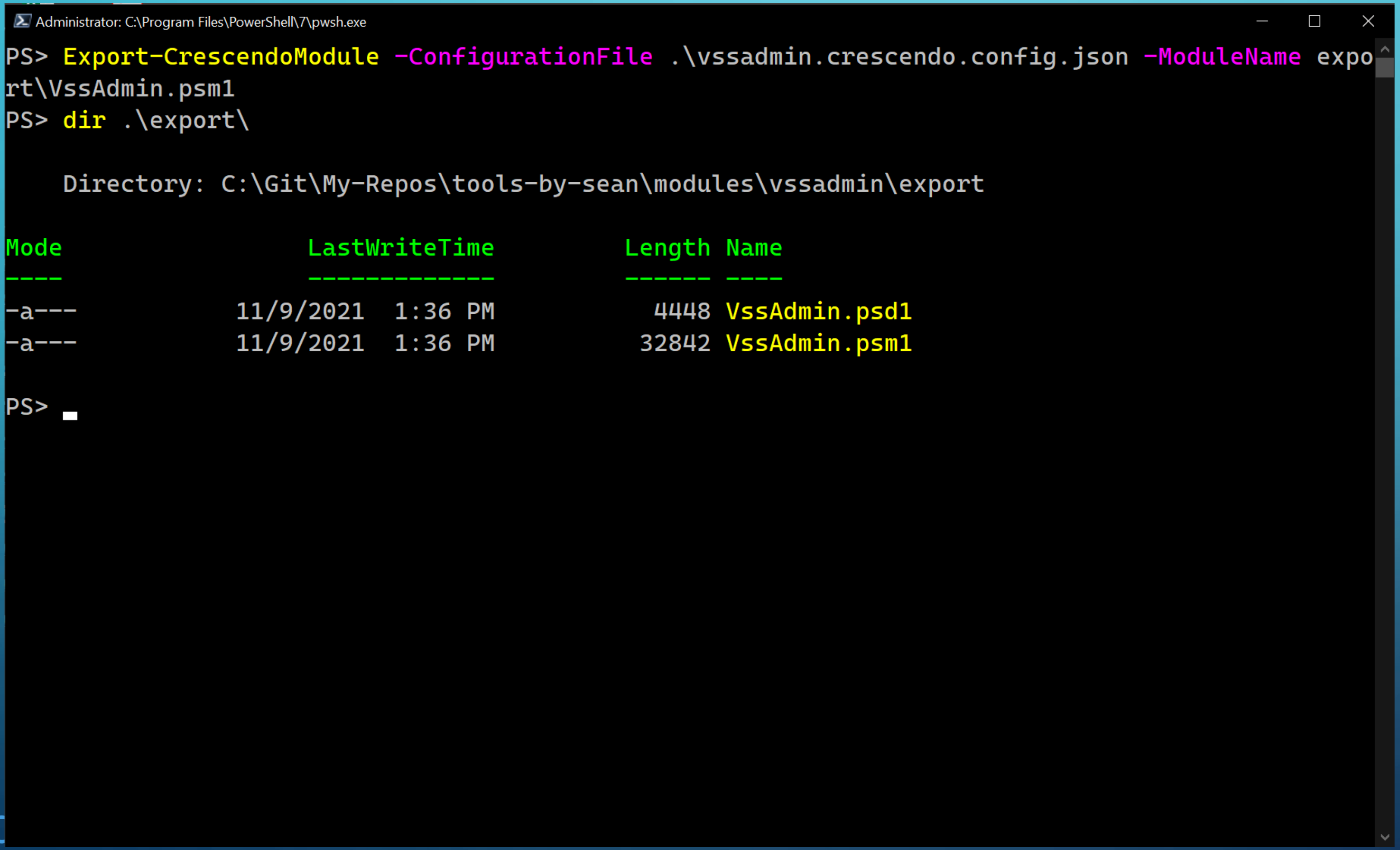
# STEP 4 - CREATE A CMDLET CONFIGURATION


File Edit Selection View Go Run Terminal Help
vssadmin.crescendo.config.json - vssadmin - Visual Studio Code
vssadmin.crescendo.config.json
vssparsers.ps1 1, M
syntax.txt 9+, M
vssadmin.crescendo.config.json > [ ] Commands > { } 2 > [ ] Parameters > { } 0
master* ⊗ 11 ⚠ 1 Docs Markdown
Ln 146, Col 57 Spaces: 4 UTF-8 CRLF JSON

```
{
    "$schema": "https://aka.ms/Crescendo/Schema.json",
    "Commands": [
        {
            "Verb": "Get",
            "Noun": "VssProvider",
            "OriginalName": "$env:Windir/system32/vssadmin.exe",
            "OriginalCommandElements": [
                "list",
                "providers"
            ],
            "Description": "List registered volume shadow copy providers",
            "Usage": {
                "Synopsis": "List registered volume shadow copy providers"
            },
            "Examples": [
                {
                    "Command": "Get-VssProvider",
                    "Description": "Get a list of VSS Providers",
                    "OriginalCommand": "vssadmin list providers"
                }
            ],
            "Platform": [
                "Windows"
            ],
            "OutputHandlers": [
```

- JSON schema helps with IntelliSense in VS Code and tooltips for help

## STEP 5 - EXPORT THE CONFIGURATION TO A MODULE



```
PS> Export-CrescendoModule -ConfigurationFile .\vssadmin.crescendo.config.json -ModuleName expo
rt\VssAdmin.psm1
PS> dir .\export\

    Directory: C:\Git\My-Repos\tools-by-sean\modules\vssadmin\export

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          11/9/2021   1:36 PM           4448 VssAdmin.psd1
-a---          11/9/2021   1:36 PM          32842 VssAdmin.psm1

PS>
```

- Crescendo creates the module files
- The end-user does not need Crescendo

## WHAT'S NEXT

- New features in Preview 4
  - Helper cmdlets
  - Experimental Help parsers
  - Improved documentation
- GA targeted for early 2022

## GET STARTED

- Get Crescendo
  - https://www.powershellgallery.com/packages/Microsoft.PowerShell.Crescendo
- Give us feedback
  - https://github.com/PowerShell/Crescendo
- Crescendo Documentation
  - https://docs.microsoft.com/powershell/module/microsoft.powershell.crescendo/

## RESOURCES

- Crescendo announcements - https://devblogs.microsoft.com/powershell/tag/powershell-crescendo/
- Sean’s 4-part blog series - https://devblogs.microsoft.com/powershell-community/tag/crescendo/
- Sean’s VSSAdmin examples - https://github.com/sdwheeler/tools-by-sean/tree/master/modules/vssadmin