INTRODUCTION & BASICS OF WINDOWS POWERSHELL

What is PowerShell?

- A task-based command-line shell and scripting language built on the .NET Framework
- Helps users control and automate the administration of the Windows operating system and applications that run on Windows

What are the features of PowerShell?

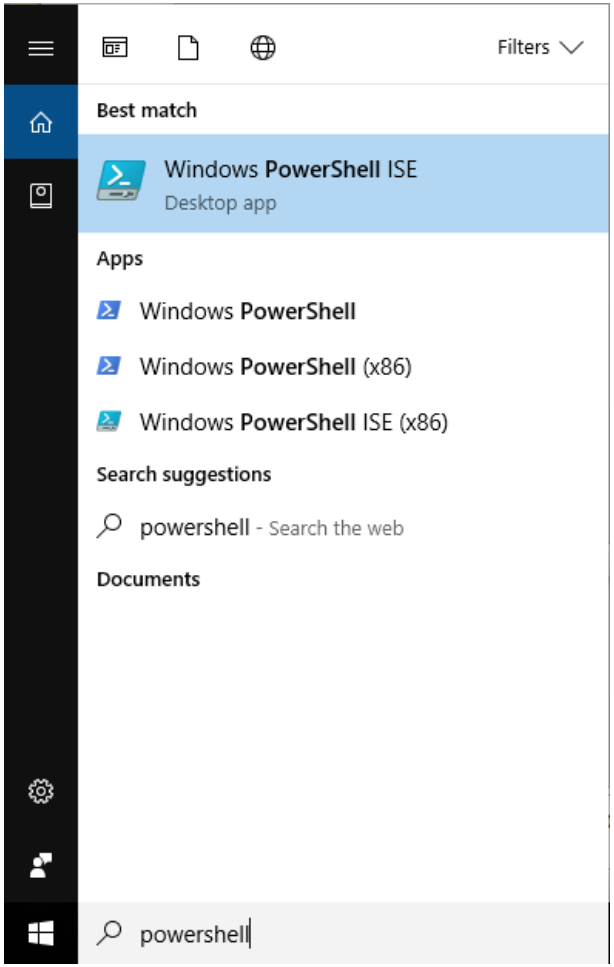
- Commands (called cmdlets) for performing common system administration tasks, such as managing the registry, services, processes, and event logs
- It has a consistent design!
  - Because cmdlets and the system use common syntax and naming conventions, data can be easily shared
    - The output from one cmdlet can be used as the input to another cmdlet without reformatting or manipulation
- Powerful object manipulation capabilities
  - Objects can be directly manipulated or sent to other tools or databases

Why should I use PowerShell?

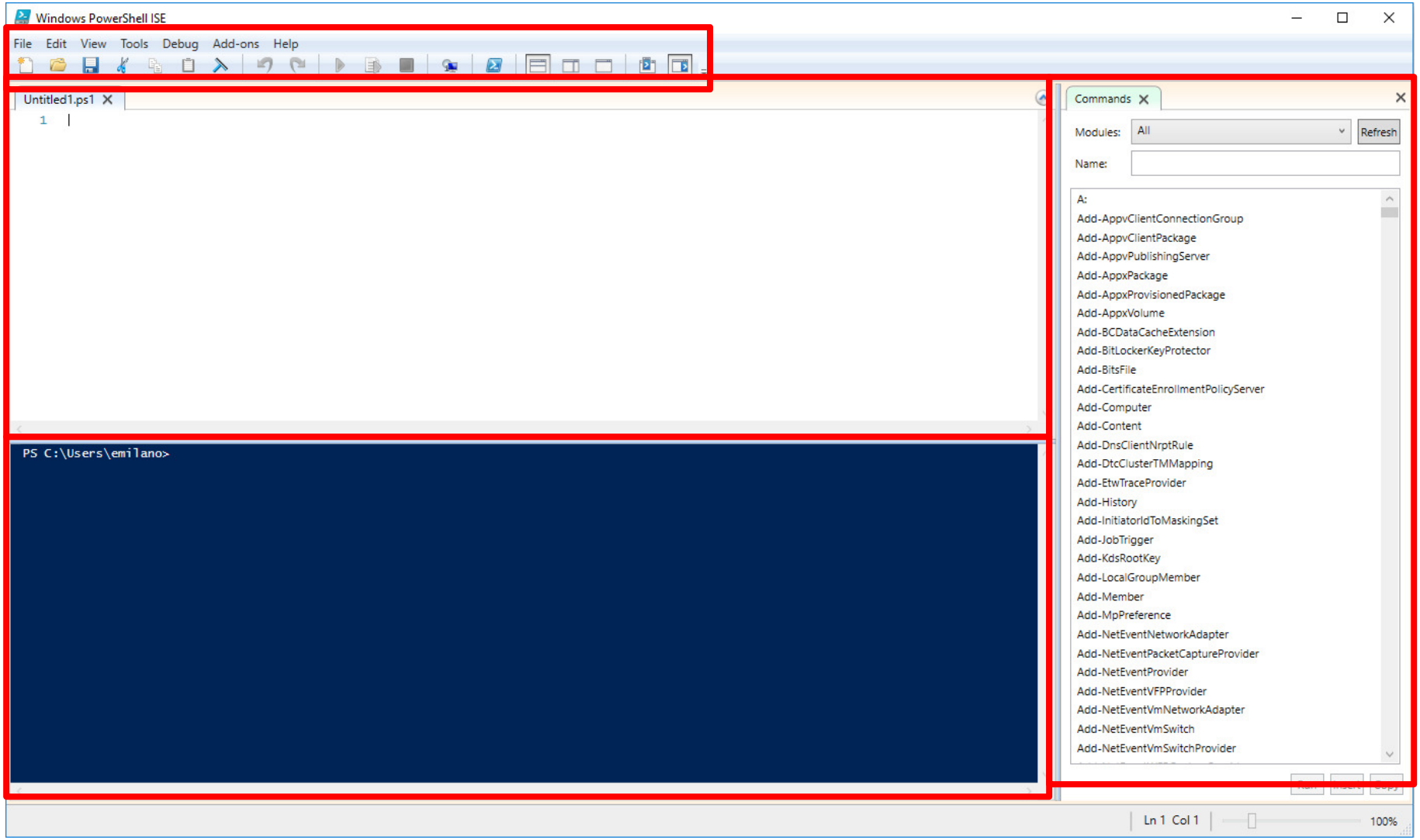
- Limit human error!
  - Repetitive tasks
    - Zipping and unzipping, printing worksheets from Excel documents
  - Processes acting on many files at once
- Automating and scheduling tasks
- Configuring Windows components and services

How can I access PowerShell?

- Standard part of Windows 7 and 8
- Can be downloaded from Microsoft Scripting Center
- Once installed, can be accessed from the Start Menu like any other application



Let's familiarize ourselves….



Familiar Faces

- # = a comment in the script
- $ = a variable in the script we define
- { } = brackets used to group statements together
- ‘ ‘ = define a string
- | = pipe, used to “pump” an output into another command

Let's walk through an example…

- Let's say I have a number of Excel files that needed to have a particular cell updated to a new value (perhaps "Revenue" in cell C3 was misspelled).
- Let's take a look at a script that will edit all of the Excel files in the specified folder and change the value of that cell…

Let's walk through an example…

Untitled1.ps1 | Example 1 Script_Excel Updates.ps1

```powershell
$files = Get-ChildItem 'C:\Users\emilano\Desktop\HFMA\Excel Test'

$xl = new-object -c excel.application

foreach($file in $files)

{

$WorkBook = $xl.Workbooks.Open($file.Fullname)
$WorkSheets = $WorkBook.WorkSheets | where {$_.name -eq "Sheet1"}
$Worksheets.Cells(3,3).Value = "Revenue"
$Workbook.close($true) #save and close workbook

}

$xl.quit()
```

Now let's write one together!

- Say I received six months worth of files from multiple clients. For example, a listing of account numbers and NDCs for outpatient Medicaid claims billed with >= $5,000 in drug charges
- I need to reorganize them by month so that I can use the appropriate fee schedule to price the claims.
- The files are big and my manager, who works off site, needs only the January files emailed to him.

Want more?

- Microsoft Virtual Academy | Getting Started with Microsoft PowerShell
- Microsoft Developer Network | Powershell