



Improve PowerShell Code Efficiency with .NET Runspaces

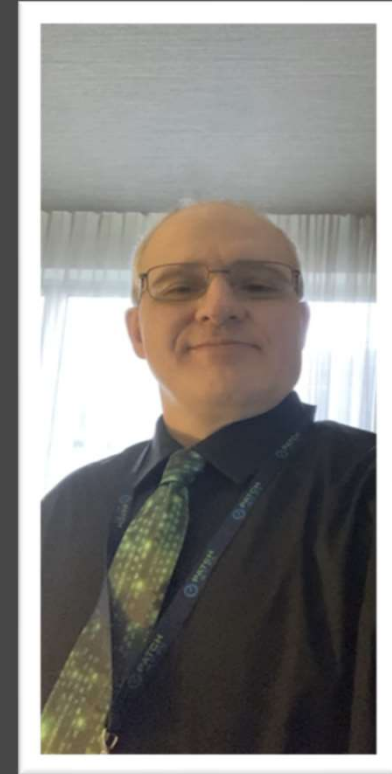
Bruce Stump

April 8-11, 2024

1 / 37

Who Am I

Everwise Credit Union
Systems Administrator
App Xtender/Enhancer
SMA OpCon
PowerShell



Everwise Credit Union



GitHub



X



LinkedIn



Documentation for Presentation (GitHub)



AGENDA

Asynchronous Processing .NET in PowerShell Runspaces Runspace Pools Demo

Why Asynchronous Processing



Faster.....



Synchronous Vs Asynchronous

Ways to create asynchronous Processes

Jobs

Reacting to Events

.Net Runspaces

ForEach-Object with –Parallel (PS 7)

Start-ThreadJob (PS 7)

.NET in PowerShell

The Following .NET Topics Covered:

Namespaces

Types

Classes

Assemblies

.NET Namespaces

Organizes types into a hierarchy, grouping types
With related functionality

e.g. [Namespace]

List .NET Namespaces

*[System.AppDomain]::CurrentDomain.GetAssemblies().
GetTypes() / Select Namespace -Unique*

.NET API Browser



April 8-11, 2024

12 / 37

.NET Types

Used to represent the functionality of an object

e.g. [Namespace.TypeName]

Classes

A set of instructions used to create instances of an object at runtime.

e.g. `[Namespace.TypeName]::ClassMethod()`

List Classes of Namespaces

```
[AppDomain]::CurrentDomain.GetAssemblies().GetTypes() |  
Where-Object {$_.NameSpace -eq 'System.IO'}
```

.NET API Browser



April 8-11, 2024

15 / 37

.NET Assemblies

A collection of Types

e.g. `[Namespace.TypeName].Assembly.location`

.NET Demo

Create a .NET RunSpace

Create a PowerShell Runspace Instance

```
[System.Management.Automation.PowerShell]  
[PowerShell]
```

```
$psInstance = [PowerShell]::Create()
```

Methods to be used with 'PowerShell' Type

AddCommand('Command')

AddParameter('ParameterName','Parameter')

AddStatement()

AddScript({Script})

Methods to be used with 'PowerShell' Type

Invoke()

BeginInvoke()

Example Runspace Creation

```
PS C:\Users\substump> [PowerShell]::Create().  
>> AddCommand( 'Get-Process' ).  
>> AddParameter( 'Name', 'pwsh' ).  
>> Invoke()
```

```
PS C:\Users\substump> $script = @"  
>> Get-Service  
>> "@  
PS C:\Users\substump> $psInstance2 = [PowerShell]::Create().AddScript($script).Invoke()  
PS C:\Users\substump> $psInstance2
```

Create a .NET Runspace Pool

Create a PowerShell Runspace Pool Instance

```
$rsp = [RunspaceFactory]::CreateRunspacePool(1,5)  
  
$rsp.Open()  
  
$rs.RunspacePool = $rsp
```


PowerShell Runspace – Deallocate Objects

`$jb.Instance | Foreach-Object Dispose`

`$rsp.Dispose()`

`[GC]::Collect()`

Example Runspace Pool

```
# Create RunspacePool, only complete 4 at a time.
$rsp = [RunspaceFactory]::CreateRunspacePool(1,4)
# Open RunspacePool.
$rsp.Open()
$jb = 1..20 | % {
    $inst = [PowerShell]::Create().AddScript('Start-Sleep -s 20')
    # Assign RunspacePool to each Runspace.
    $inst.RunspacePool = $rsp
    [PSCustomObject] @{
        Id = $inst.InstanceId
        Instance = $inst
        AsyncResult = $inst.BeginInvoke()
        # The InvocationStateInfo Property is used to show if object is running.
    } | Add-Member State -MemberType ScriptProperty -PassThru -Value {
        $this.Instance.InvocationStateInfo.State
    }
}
```

.NET Runspace Demo

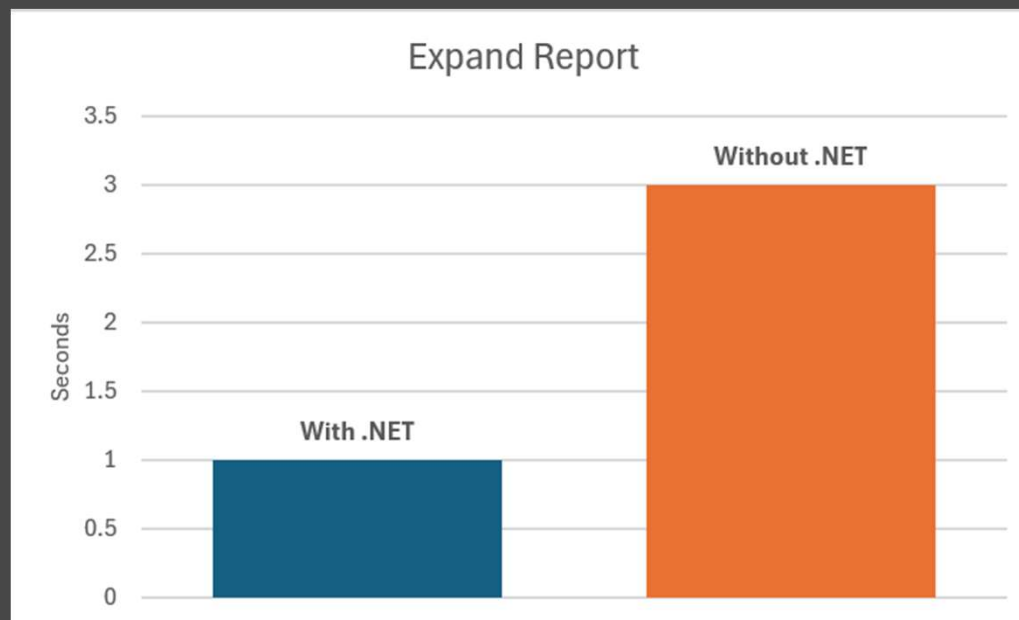
.NET Sample Code

.NET Expand Report

```
# Break up STAR file into individual reports and store in the $iReport array.  
$iReport = [PowerShell]::Create().  
    AddCommand('Expand-XtenderTextReport').  
    AddParameter('FileName', $fileToProcess).  
    AddParameter('Pattern', '^1').  
    Invoke()
```

Time to Expand Reports

1000 Reports



April 8-11, 2024

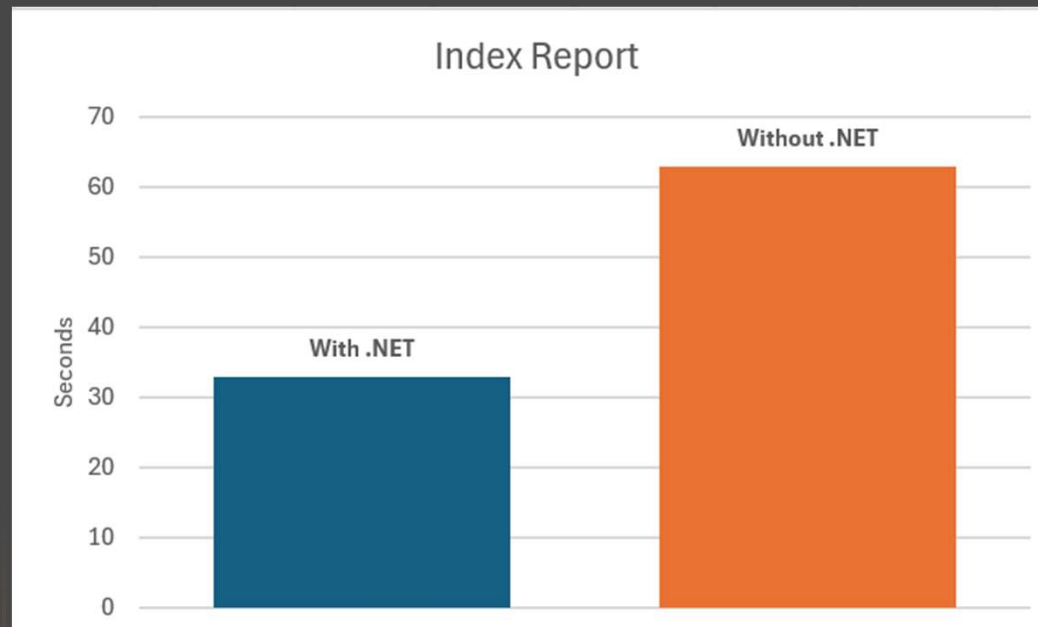
30 / 37

.NET Index Report

```
# Create Report Indexes
$iReportIndex = [PowerShell]::Create().
    AddCommand('Invoke-XtenderStarIndex').
    AddParameter('iReport',$iReport).
    AddParameter('fileToProcessPathName', $fileToProcess).
    AddParameter('ImportPath', $importPath).
    AddParameter('fileIndex',0).
    Invoke()
```

Time to Index Reports

11252 Reports



April 8-11, 2024

32 / 37

Final Demo

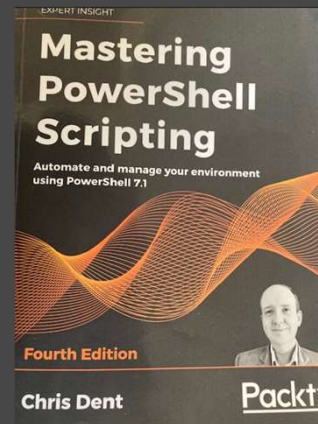
Call To Action

Good Programming Practices

Consider .NET Runspace to isolate your script

Resource

Mastering PowerShell Scripting Fourth Edition By Chris Dent



April 8-11, 2024

35 / 37



PATCH MY PC



PURE STORAGE



MANNING

Dataon



ScriptRunner
Your #1 for PowerShell Automation



April 8-11, 2014



THANK YOU

Please review this session

Documentation



LinkedIn



Session Review

April 8-11, 2024

37 / 37