BEST PRACTICES® CONFERENCE SHAREPOINT

Clarity. Direction. Confidence.

IT103: WINDOWS POWERSHELL FOR SHAREPOINT 2010 ADMINISTRATORS

Gary Lapointe, MVP

APTILLON

BEST PRACTICES CONFERENCE SHAREPOINT

About Me

- SharePoint MVP
- Independent Consultant and Owner of Falchion Consulting, LLC
  - http://www.falchionconsulting.com
- Principal Consultant - Aptillon, Inc.
  - http://www.aptillon.com
- Blog: http://blog.falchionconsulting.com/
- Twitter: @glapointe
- Email: gary@falchionconsulting.com

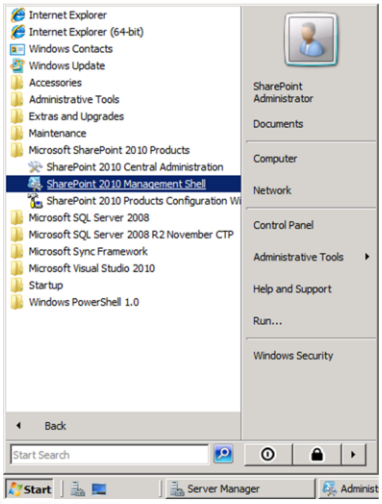
Agenda

- Windows PowerShell 101
- Common Tasks
  - Reporting
  - Maintenance
- Windows PowerShell Remoting
- Questions

WINDOWS POWERSHELL 101

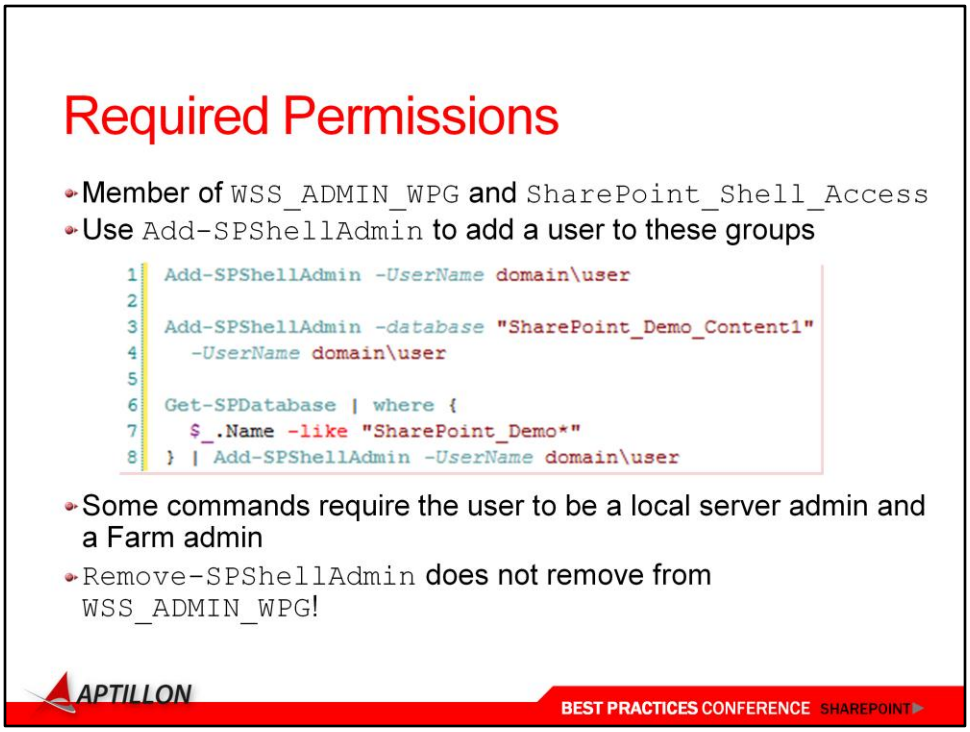
Getting on the same page...

Management Shell



- C:\Windows\System32\WindowsPowerShell\v1.0\PowerShell.exe -NoExit " & ' C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\CONFIG\POWERSHELL\Registration\sharepoint.ps1 ' "


```
if (!(Test-Path $profile.AllUsersAllHosts)) {
  New-Item -Type file -Path $profile.AllUsersAllHosts -Force
}
$sb = @"
if (`$host.UI.RawUI.WindowTitle -ne "Administrator: SharePoint 2010 Management
Shell") {
  `$ver = `$host | select version
  if (`$ver.Version.Major -gt 1) {`$host.Runspace.ThreadOptions = "ReuseThread"}
  if ((Get-PSSnapin "Microsoft.SharePoint.PowerShell" -ErrorAction SilentlyContinue) -
eq `$null) {
    Add-PSSnapin "Microsoft.SharePoint.PowerShell"
  }
}
"@
Set-Content -Path $profile.AllUsersAllHosts -Value $sb
```



SharePoint_Shell_Access role exists in the config db by default but no other databases. The Add-SPShellAdmin cmdlet will add the user to the WSS_ADMIN_WPG group on each server (not just the executing server). The Remove-SPShellAdmin cmdlet does not remove from the WSS_ADMIN_WPG group so you must manually remove the user from this group.

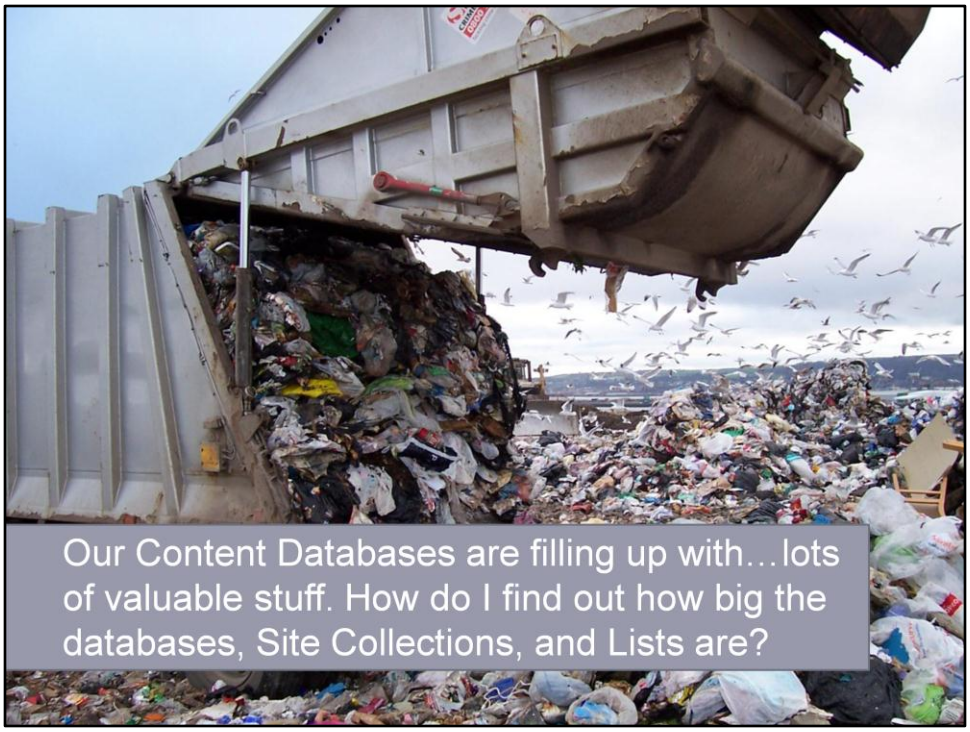
To add to the Farm Admin group:
New-SPUser -UserAlias "**domain\user**" -Web "**http://sp2010:1234/**" -Group "Farm Administrators"

DEMO…

Windows PowerShell 101

COMMON TASKS - REPORTING

Getting things done…



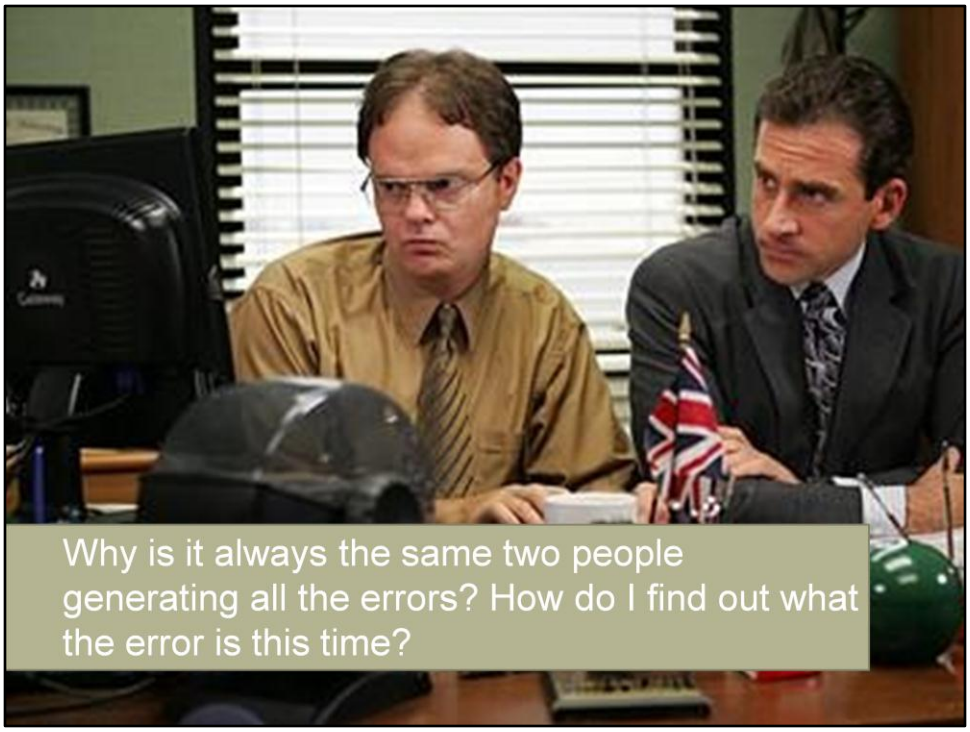

```
Get-SPDatabase | select Name, @{Expression={$_.DiskSizeRequired/1MB};Label="Size"} | sort Size -Descending

get-spsite -limit all | select url, @{Expression={$_.Usage.Storage/1MB};Label="Size"},@{Expression={$_.AllWebs.Count};Label="Webs"} | sort Size -Descending

Get-SPWeb -Site http://demo/sites/b20 | % {$_.Lists} | select @{Expression={$_.RootFolder.ServerRelativeUrl};Label="Url"},Title,ItemCount | sort ItemCount
```


```
function Get-DocInventory() {
   foreach ($site in (Get-SPSite -Limit All)) {
      foreach ($web in $site.AllWebs) {
         foreach ($list in $web.Lists) {
            if ($list.BaseType -ne "DocumentLibrary") {
               continue
            }
            foreach ($item in $list.Items) {
               $data = @{
                  "Site" = $site.Url
                  "Web" = $web.Url
                  "list" = $list.Title
                  "Item ID" = $item.ID
                  "Item URL" = $item.Url
                  "Item Title" = $item.Title
                  "Item Created" = $item["Created"]
                  "Item Modified" = $item["Modified"]
                  "File Size" = $item.File.Length/1KB
               }
               New-Object PSObject -Property $data
            }
         }
         $web.Dispose();
      }
      $site.Dispose()
   }
}
Get-DocInventory | Out-GridView
#Get-DocInventory | Export-Csv -NoTypeInformation -Path c:\inventory.csv
```

Get-SPUserEffectivePermissions.ps1

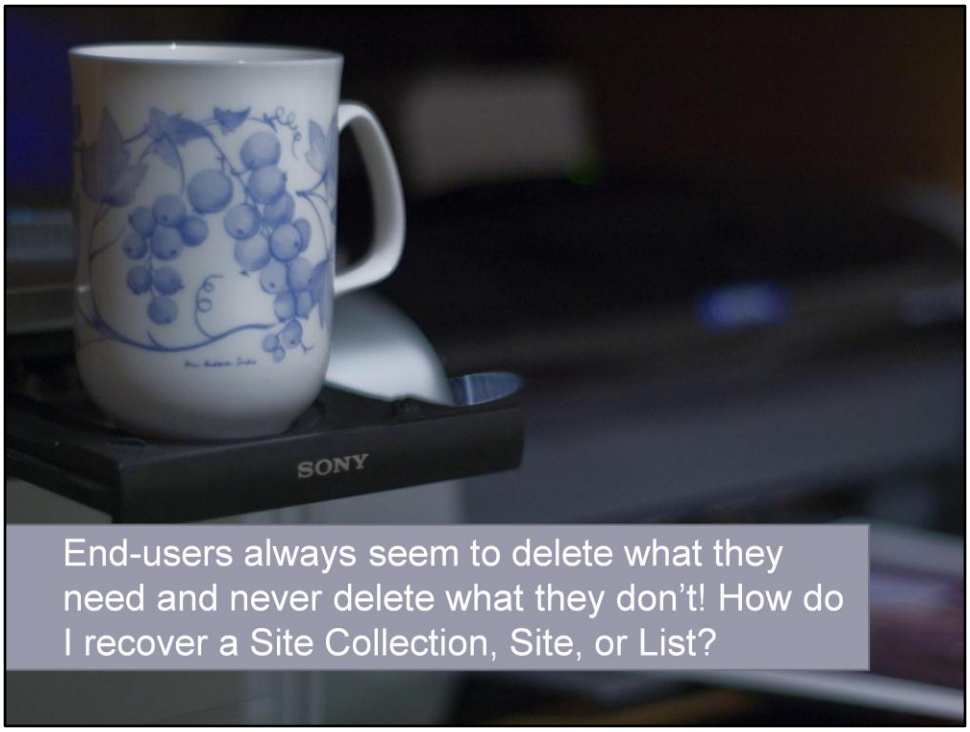
COMMON TASKS - MAINTENANCE

Getting things done…

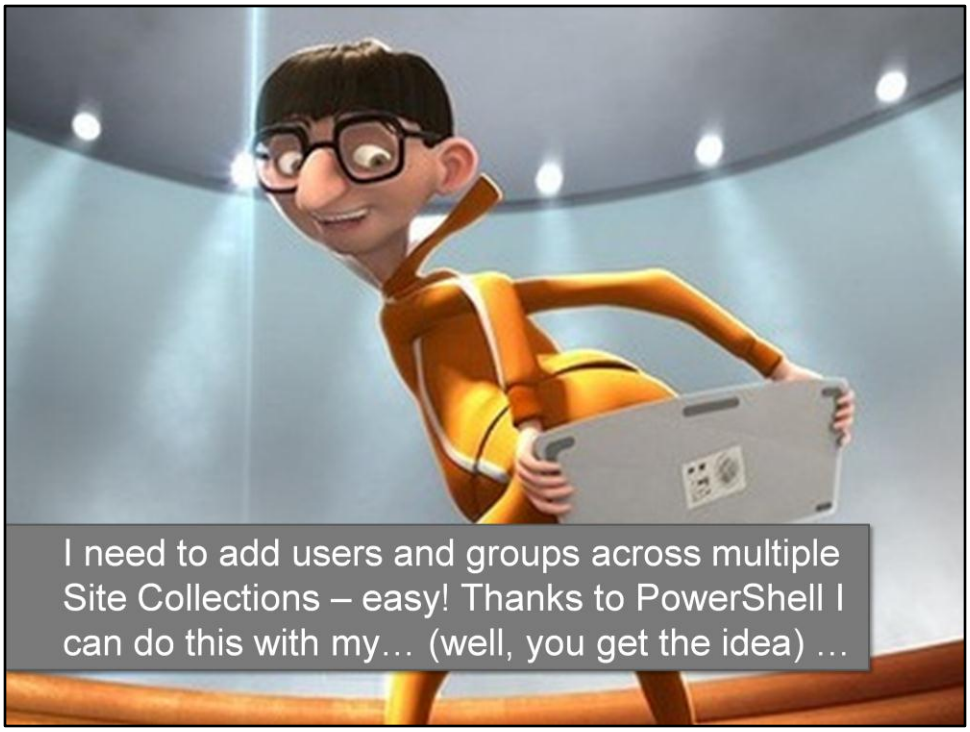



```
Get-SPLogEvent -StartTime (Get-Date).AddMinutes(-10) | ? {$_.Correlation -eq "8db5e7ed-075c-46cc-8d7c-e2cb78f15f7e"}
```

```
Merge-SPLogFile -Path c:\events.log -StartTime (Get-Date).AddMinutes(-10) -EndTime (Get-Date) -Correlation "e7ed36e1-4966-4f91-b3cc-3951ba4c5e38"
```




```
$db = Get-SPContentDatabase -ConnectAsUnattachedDatabase -DatabaseName "SharePoint_Restore1" -DatabaseServer "spsql1"

$restoredSite = $db | Get-SPSite | where { $_.ServerRelativeUrl -eq "/sites/bpcuk" }
Backup-SPSite -Identity $restoredSite -Path C:\backups\Demo.bak -NoSiteLock

$web = $restoredSite | Get-SPWeb "/sites/bpcuk/ps4admins"
Export-SPWeb -Identity $web -Path C:\ExportData\PS4Admins -NoFileCompression -IncludeVersions All -IncludeUserSecurity


new-spweb -Url http://demo/sites/bpcuk/PS4Admins -AddToTopNav -Name "PowerShell for Administrators"
import-spweb -Identity http://demo/sites/bpcuk/ps4admins -Path C:\ExportData\PS4Admins -NoFileCompression -IncludeUserSecurity
```




```
$user = new-spuser -UserAlias aptillon\glapointe -Web http://demo
$group = Get-SPGroup -web http://demo -group "Demo Members"
$group.AddUser($user)

$group = New-SPGroup -web http://demo -groupName "BPC Group" -ownerName
aptillon\spadmin -desc "BPC Group" -memberName aptillon\spadmin

Add-SPGroupPermission -group $group -permissions "Contribute","Design"
```

WINDOWS POWERSHELL REMOTING

Administering SharePoint remotely…

Remoting

- PowerShell Remoting uses WinRM, Microsoft's implementation of the WS-Management protocol
  - WinRM allows you to run scripts against remote servers over HTTP and HTTPS
- Works with V2 only
- Requires that WinRM be enabled on both the client and the server

Enabling Remoting

- Run `Enable-PsRemoting` on the client and server machines
- Must Enable CredSSP
  - Credential Security Support Provider
  - Allows cmdlets to talk to SQL using the provided credentials (handles the double-hop issue)
- Increase the `MaxMemoryPerShellMB` setting on remote server (default is 150MB)
  - `Set-Item WSMan:\localhost\Shell\MaxMemoryPerShellMB 1000`
- Decrease `MaxShellsPerUser` and `MaxConcurrentUsers` (default is 5)
  - `Set-Item WSMan:\localhost\shell\MaxShellsPerUser 2`
  - `Set-Item WSMan:\localhost\shell\MaxConcurrentUsers 2`

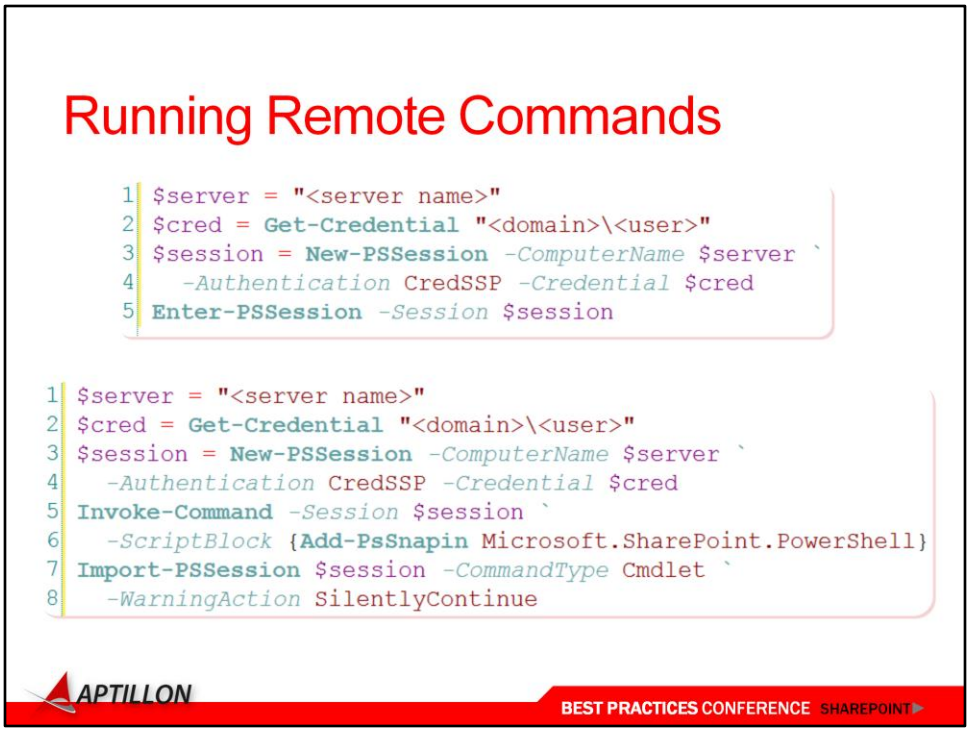
Enabling CredSSP

- On the client machine
  - Group Policy must be edited to allow credential delegation to the target computer.
    - Use gpedit.msc

      Computer Configuration -> Administrative Templates -> System -> Credentials Delegation -> Allow Delegating Fresh Credentials

      Verify that it is enabled and configured with an SPN appropriate for the target computer (WSMAN/myserver.domain.com or WSMAN/*.domain.com)

      May have to similarly enable “Allow Fresh Credentials with NTLM-only Server Authentication” if the above setting does not work
  - `Enable-WSmanCredSSP -Role Client -DelegateComputer <remote server name>`
- On the server machine
  - `Enable-WSmanCredSSP -Role Server`



$server = "sp2010"
$cred = Get-Credential aptillon\spadmin
$session = New-PSSession -ComputerName $server -Authentication CredSSP -Credential $cred

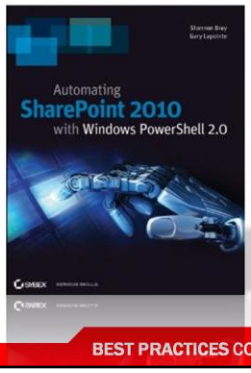
This:
Enter-PSSession -Session $session
Or this:
Invoke-Command -Session $session -ScriptBlock {Add-PsSnapin Microsoft.SharePoint.PowerShell}
Import-PSSession $session -CommandType Cmdlet -WarningAction SilentlyContinue

DEMO…

Windows PowerShell Remoting

Key Takeaways

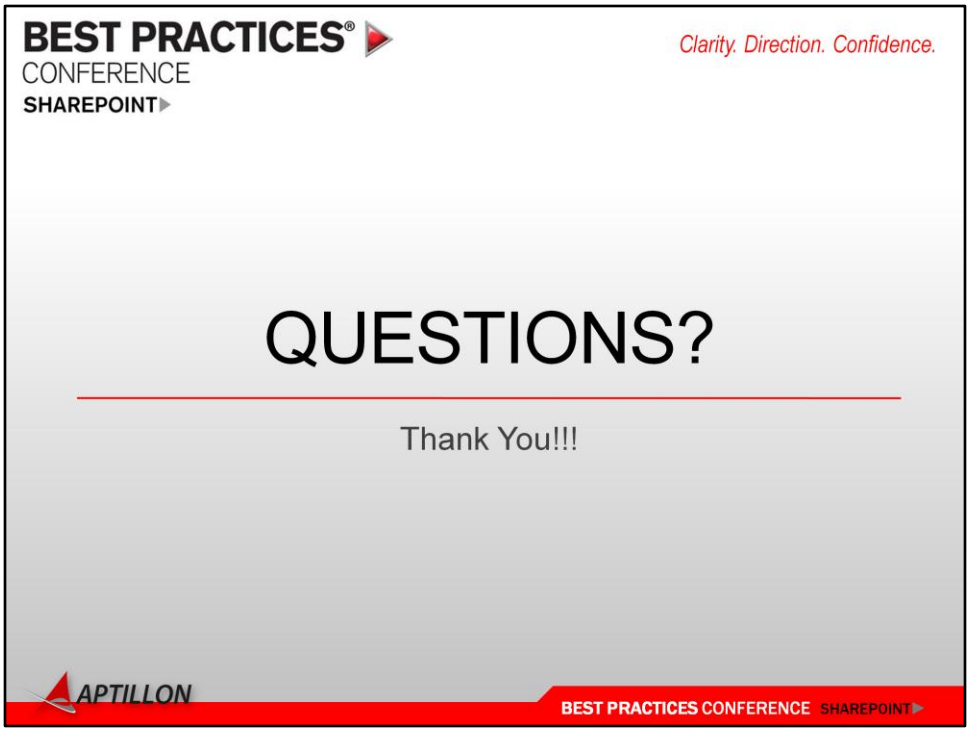
- Learn PowerShell!!!
- Get comfortable with the SharePoint API – it's not just for developers anymore!!!
- Buy my book!!!



About Aptillon

- SharePoint MVPs
- Microsoft Certified Master
- Consultants, Trainers, Authors, Speakers, Bloggers
- Great People, Great Skill, Great Talent, Great Passion

Todd Baginski | David Mann | Gary Lapointe | Darrin Bishop | Maurice Prather | Dan Holme | Matthew McDermott


BEST PRACTICES®
CONFERENCE
SHAREPOINT
Clarity. Direction. Confidence.
QUESTIONS?
Thank You!!!
APTILLON
BEST PRACTICES CONFERENCE SHAREPOINT