Cherwell + PowerShell = Powerful Workflow Automation

Robert Goguen works as the Change Manager and Senior IT Operations Analyst for Irving Oil Limited, which operates Canada's largest oil refinery. An IT veteran with 20 years experience, half of his career has been in the IT learning development and performance field. Goguen has extensive experience in ITSM, and has been a member of the core ITIL implementation teams for Incident, Change, Problem and Configuration Management. As a subject matter expert in process implementation and development, he has been the lead on Cherwell Service Management development, administration, and operational strategy. Goguen has also overseen the development and deployment of training for Cherwell Service Management, HEAT, Axios, Remedy, SalesForce, Oracle BI, and SDLC.

Jeff Jones of Excalibur Data Systems is a veteran of multiple ITSM toolsets, just about every programming language known to man and way too much digital technology. He has spent the past 25+ years working with customers to improve their processes, integrate their data and when needed code things into submission. He has been working with Cherwell Service Management since its earliest days on the market and is a Cherwell consultant and instructor. When he isn't working with a customer to make their Cherwell System leap tall building in a single bound, you'll probably find him learning yet another programming language or elbows deep in a new technology he wants to add to his tool bag.

Cherwell GLOBAL CONFERENCE

Discovering the Nature of Superior Service

Cherwell + PowerShell = Powerful Workflow Automation

Robert Goguen
Service Management, Irving Oil Limited

Jeff Jones
Consultant, Excalibur Data Systems

www.cherwellgc.com #CGC15

Agenda

- Welcome & Introduction
- About this Session
- PowerShell
  - What is PowerShell?
  - What can I do with PowerShell?
  - Why PowerShell?
- Other scripting languages & non-Windows devices
- How the magic happens
  - Executing Scripts
  - Passing Variables
  - Working with the Output
  - Enviroment Considerations (SaaS, On-Premise, O365)
  - Security & Privileges
- Live Demo

Cherwell + PowerShell = Powerful Workflow Automation

Your Mission…..

- Increase Value to the Business
- Reduce Cost
- Minimize Risk
- Improve Service Offerings
- No impact on Quality or Services

PowerShell

- What is PowerShell?

PowerShell is an automation platform and scripting language for Windows and Windows Server that allows you to simplify the management of your systems. Unlike other text-based shells, PowerShell harnesses the power of the .NET Framework, providing rich objects and a massive set of built-in functionality for taking control of your Windows environments (Source: MSDN)

- Why PowerShell?

Most Microsoft Systems are managed using PowerShell. It is Microsoft's go-forward technology that will be used to manage all products in the future. (Currently on version 5.0)

PowerShell

```
Cmdlet          Set-PSBreakpoint                            Microsoft.PowerShell.Utility
Cmdlet          Set-PSDebug                                 Microsoft.PowerShell.Core
Cmdlet          Set-PSSessionConfiguration                  Microsoft.PowerShell.Core
Cmdlet          Set-ScheduledJob                            PSScheduledJob
Cmdlet          Set-ScheduledJobOption                      PSScheduledJob
Cmdlet          Set-Service                                 Microsoft.PowerShell.Management
Cmdlet          Set-StrictMode                              Microsoft.PowerShell.Core
Cmdlet          Set-TraceSource                             Microsoft.PowerShell.Utility
Cmdlet          Set-Variable                                Microsoft.PowerShell.Utility
Cmdlet          Set-WmiInstance                             Microsoft.PowerShell.Management
Cmdlet          Set-WSManInstance                           Microsoft.WSMan.Management
Cmdlet          Set-WSManQuickConfig                        Microsoft.WSMan.Management
Cmdlet          Show-Command                                Microsoft.PowerShell.Utility
Cmdlet          Show-ControlPanelItem                       Microsoft.PowerShell.Management
Cmdlet          Show-EventLog                               Microsoft.PowerShell.Management
Cmdlet          Sort-Object                                 Microsoft.PowerShell.Utility
Cmdlet          Split-Path                                  Microsoft.PowerShell.Management
Cmdlet          Start-BitsTransfer                          BitsTransfer
Cmdlet          Start-Job                                   Microsoft.PowerShell.Core
Cmdlet          Start-Process                               Microsoft.PowerShell.Management
Cmdlet          Start-Service                               Microsoft.PowerShell.Management
Cmdlet          Start-Sleep                                 Microsoft.PowerShell.Utility
Cmdlet          Start-Transaction                           Microsoft.PowerShell.Management
Cmdlet          Start-Transcript                            Microsoft.PowerShell.Host
Cmdlet          Stop-Computer                               Microsoft.PowerShell.Management
Cmdlet          Stop-Job                                    Microsoft.PowerShell.Core
Cmdlet          Stop-Process                                Microsoft.PowerShell.Management
Cmdlet          Stop-Service                                Microsoft.PowerShell.Management
Cmdlet          Stop-Transcript                             Microsoft.PowerShell.Host
Cmdlet          Suspend-BitsTransfer                        BitsTransfer
Cmdlet          Suspend-Job                                 Microsoft.PowerShell.Core
Cmdlet          Suspend-Service                             Microsoft.PowerShell.Management
Cmdlet          Tee-Object                                  Microsoft.PowerShell.Utility
Cmdlet          Test-AppLockerPolicy                        AppLocker
Cmdlet          Test-ComputerSecureChannel                  Microsoft.PowerShell.Management
Cmdlet          Test-Connection                             Microsoft.PowerShell.Management
Cmdlet          Test-ModuleManifest                         Microsoft.PowerShell.Core
Cmdlet          Test-Path                                   Microsoft.PowerShell.Management
Cmdlet          Test-PSSessionConfigurationFile             Microsoft.PowerShell.Core
Cmdlet          Test-WSMan                                  Microsoft.WSMan.Management
```

What can I do with PowerShell?

Just About Anything!

100’s of Cmdlet’s, Functions and 1000’s of Parameters

Other scripting languages & non-Windows devices

UNIX®

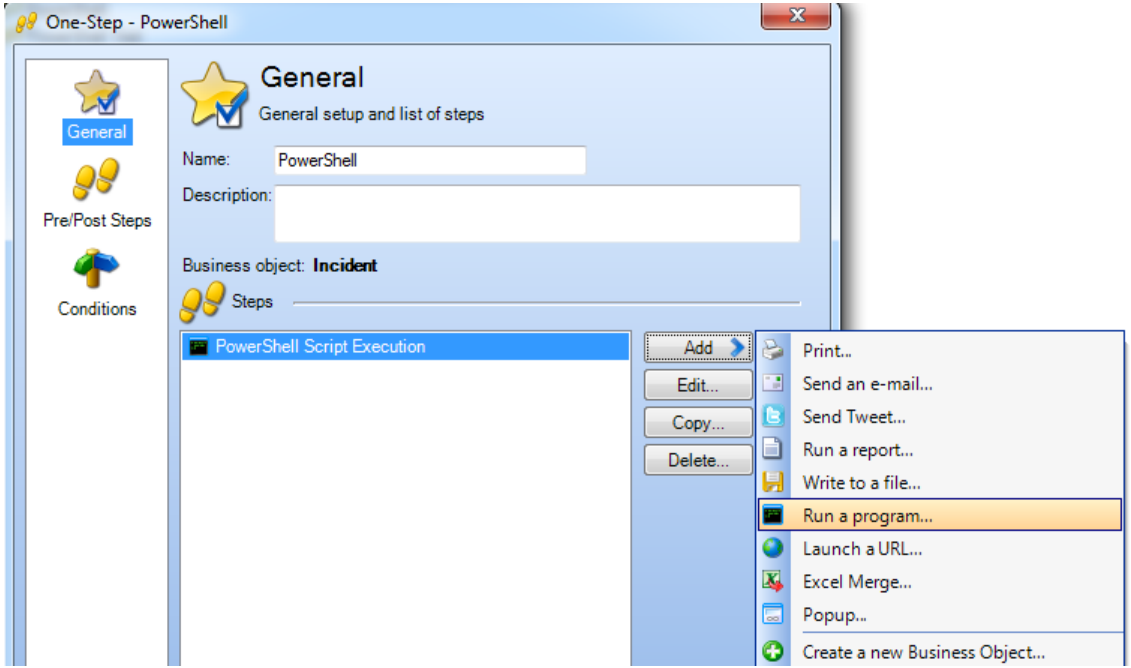
- What if I have to use a different scripting language?

This technique is equally effective for other scripting language on Windows operating systems

- What if I need to script for a non-Windows system or device?

Use PowerShell's new SSH features to run a remote script via SSH in whatever scripting language the system or device requires.

How the Magic Happens – Executing Scripts



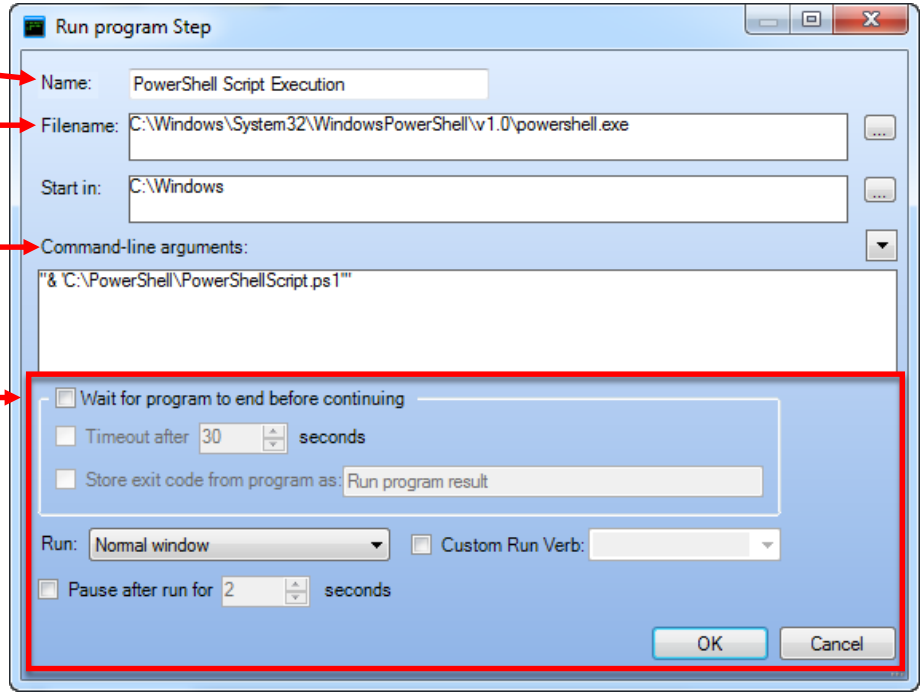
How the Magic Happens – Executing Scripts

Display Name of Action

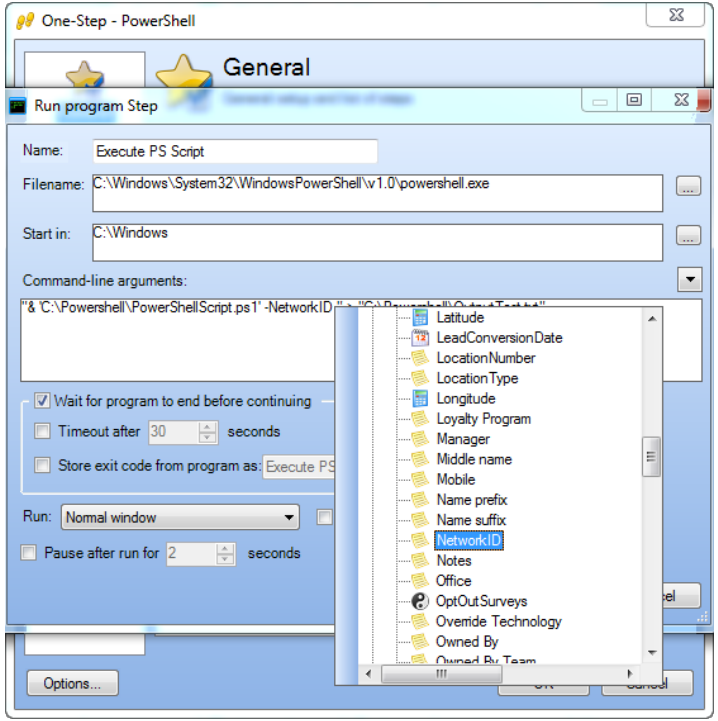
Location of PowerShell Executable
(C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe)

Execution of Script with parameters
(“& ‘script location\scriptname.ps1’ ”)

Define how One-Step Interacts with Script Execution

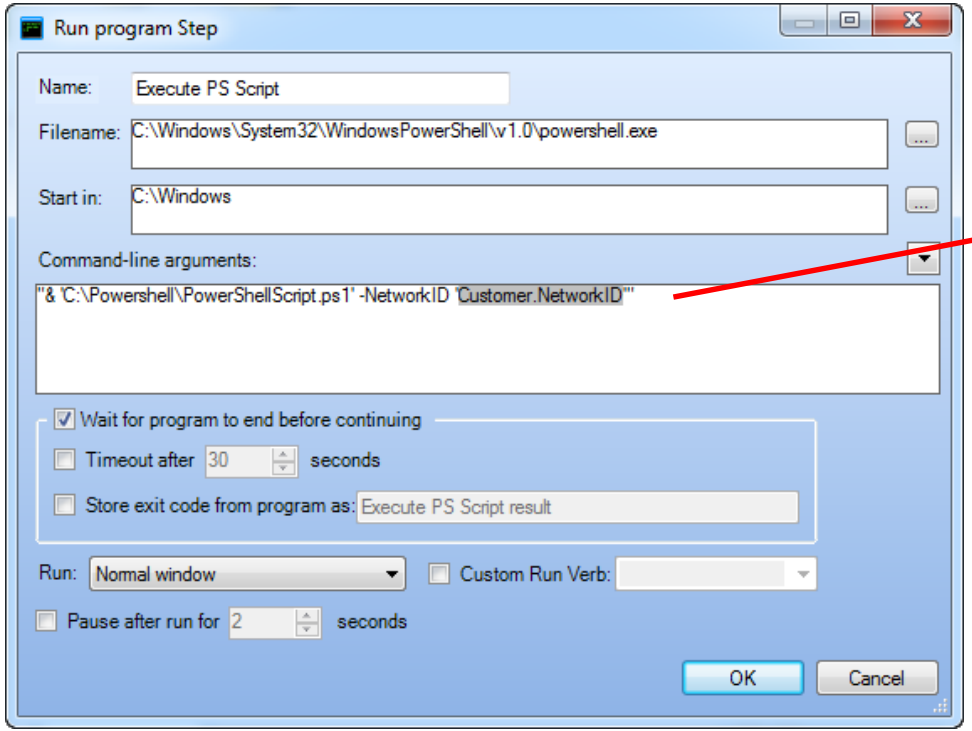


How the Magic Happens – Passing Variables



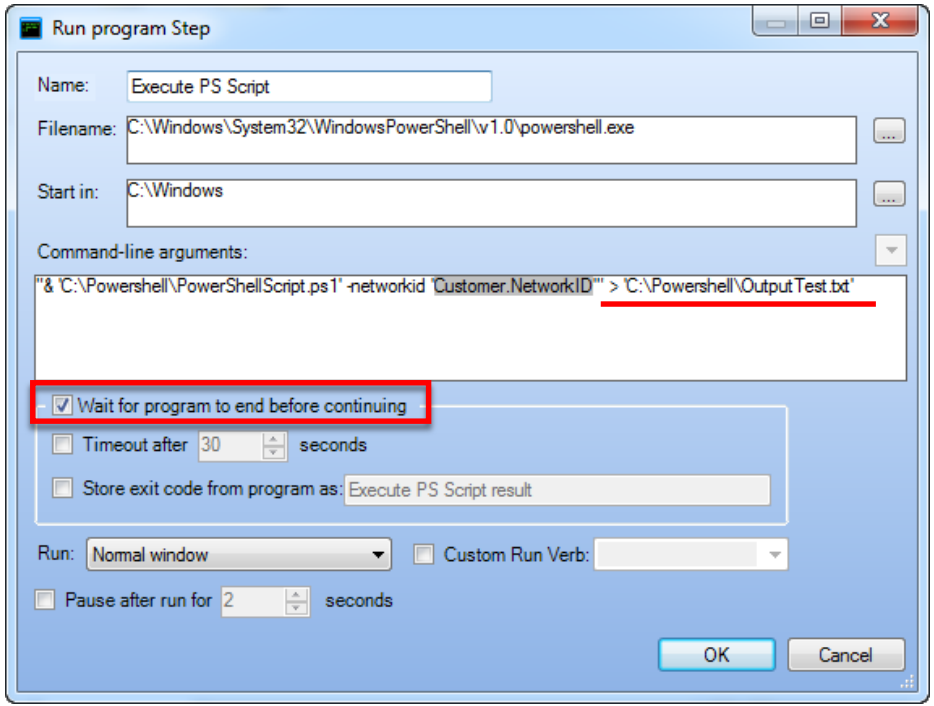
"& 'Script L*ocation' –Parameter1 Value"*

How the Magic Happens – Passing Variables



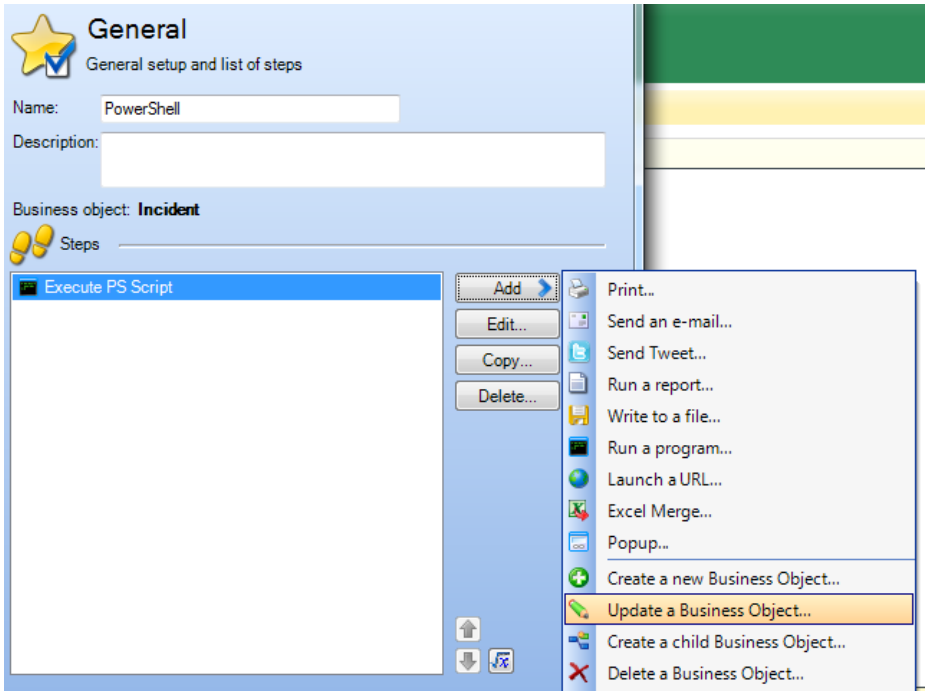

```
1  #Command Line argument handling
2  Param(
3      [string]$networkid
4  )
5
6  Get-ADUser $networkid
7
```

How the Magic Happens – Working with Output

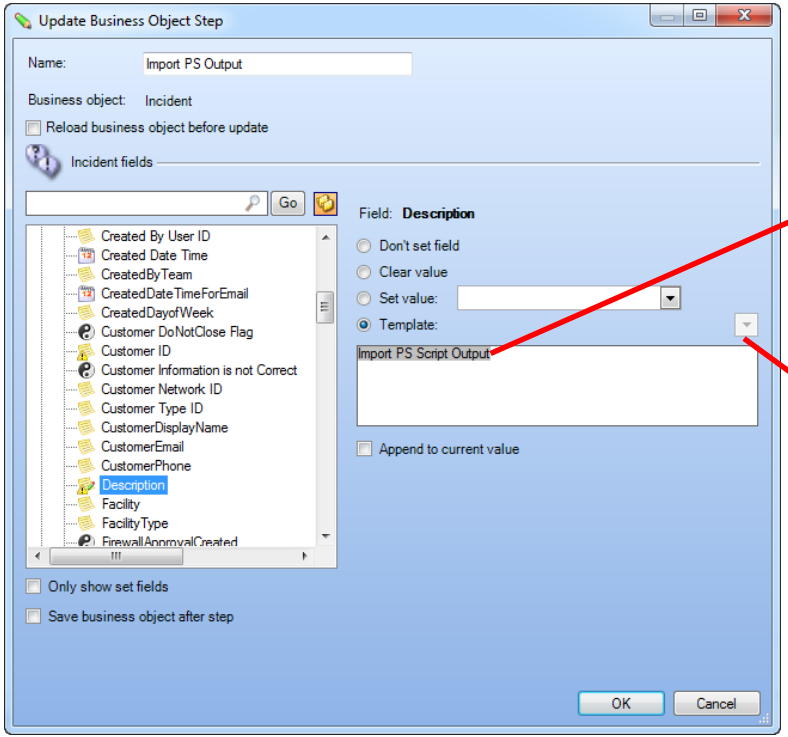


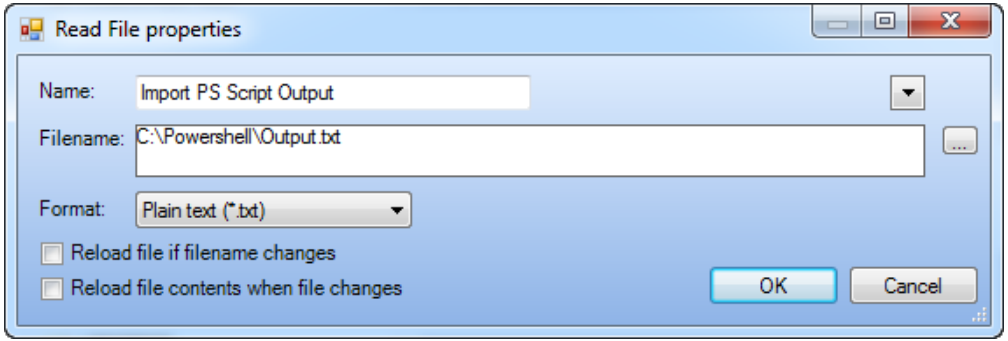
> “<location>\Filename.txt”

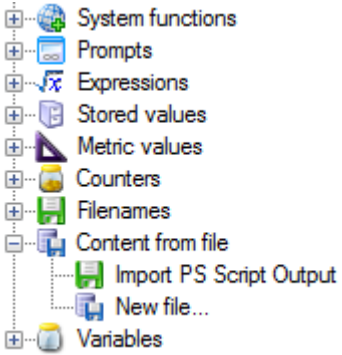
How the Magic Happens – Working with Output



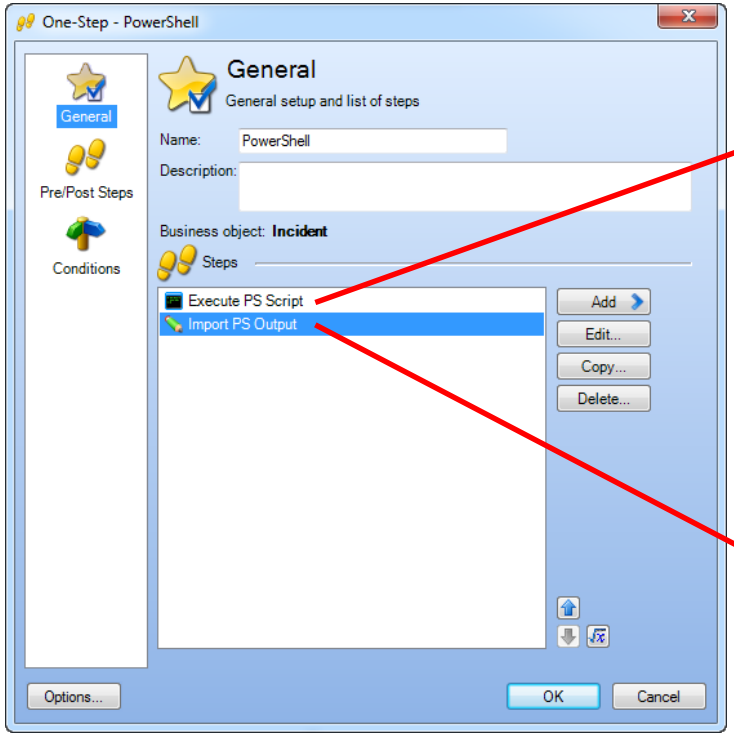
How the Magic Happens – Working with Output

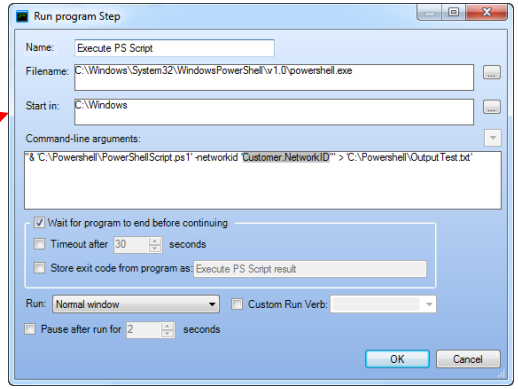


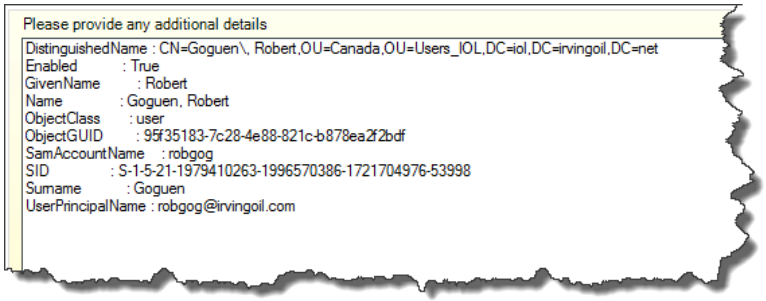




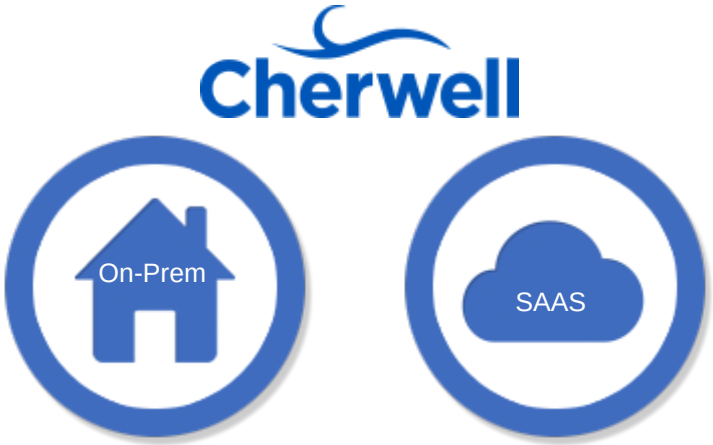
How the Magic Happens – Working with Output







How the Magic Happens- Environment Considerations



The Key is Visibility

How the Magic Happens – Security & Privileges

Keep in mind that these scripts run in the security context of the user executing them.

This means that for users executing the One-Steps through the rich client the script will execute in their security context.

For scheduled jobs they run in the security context of the service account used to run the Cherwell Scheduling Server.

Make sure the scripts don't require more security privileges than the user can provide.


DEMO

Additional Resources

Our Presentation Resource Website
http://www.excaliburdata.com/resources/

Microsoft PowerShell for O365 Website
http://powershell.office.com/

MSDN PowerShell Gallery
https://msdn.microsoft.com/powershell

Any Questions?


Cherwell
GLOBAL
CONFERENCE
Discovering the Nature of Superior Service
Thank you for attending this session.
Please fill out an evaluation form.
www.cherwellgc.com
#CGC15