PowerShell for Exchange Admins

- Get-Speaker | FL

Name : Kamal Abburi
Title : Premier Field Engineer
Expertise : Exchange
Email : Kamal.Abburi@Microsoft.com
Blog : mrproactive.com

Note: Inspired by my fellow PFEs

What We Do

Proactive Services
- Workshops
- Health Checks
- Risk Assessments
- Supportability Reviews
- Chalk & Talks
- Knowledge Transfers

Reactive Support

Onsite and Remote

Global Community

Troubleshooting & Root Cause Analysis

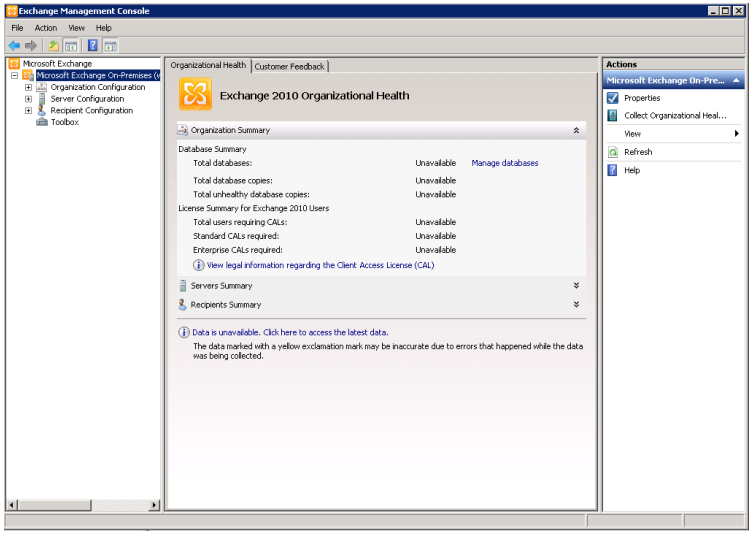
Technical Leadership

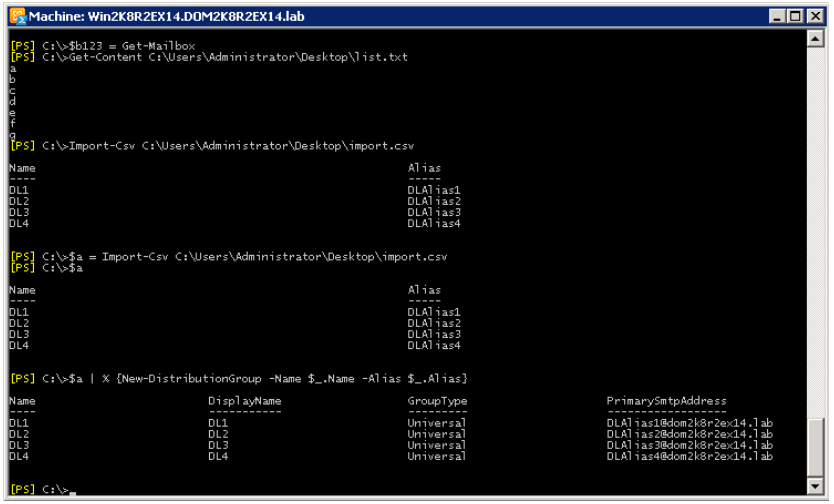
Partner with Product Groups

Overview

- PowerShell
- Exchange and PowerShell
- Tips and Tricks

Architecture





PowerShell Engine

Exchange Cmdlets

AD

Registry

MAPI

Metabase

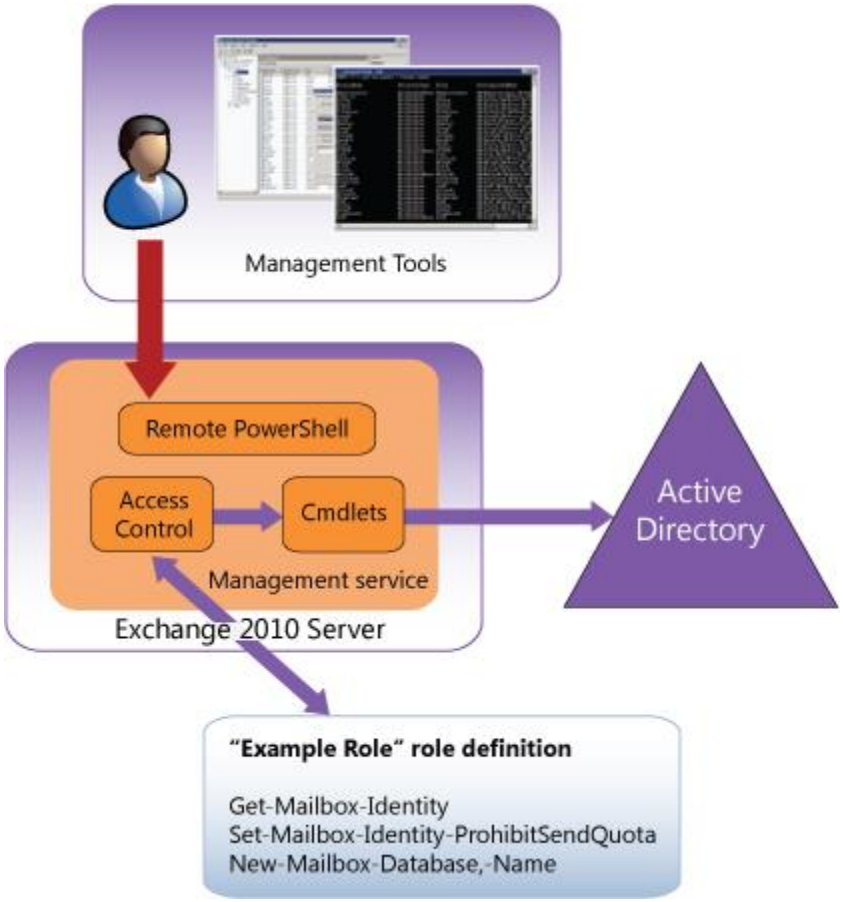
What is PowerShell

- Command Driven Environment
- A scripting language
- Based on .NET Framework
- Unit of operation is a cmdlet
- Cmdlets are .NET classes
- All Exchange management operations are implemented as PowerShell cmdlets

Exchange Management Shell

- Built upon PowerShell technology
- Windows Management Framework
- PowerShell SDK available
- Role Based Access Control – Exchange 2010

RBAC and the Shell



Source : Technet

PowerShell Commands in the EMC

- Logs every Shell command in the EMC

View Exchange Management Shell Command Log<Exchange Server 2010>

File Action

Start Execution Time	End Execution Time	Execution Status	Command
9/11/2012 4:46:47 PM	9/11/2012 4:46:49 PM	Completed	Get-ExchangeServer| Filter-...
9/11/2012 4:46:50 PM	9/11/2012 4:46:51 PM	Completed	Get-OrganizationConfig
9/11/2012 4:46:57 PM	9/11/2012 4:46:57 PM	Completed	Get-Recipient -PropertySet ...
9/11/2012 5:51:31 PM	9/11/2012 5:51:31 PM	Completed	Get-UserPrincipalNamesSu...
9/11/2012 5:51:33 PM	9/11/2012 5:51:34 PM	Completed	Get-Mailbox -Identity 'DOM...
9/11/2012 5:51:34 PM	9/11/2012 5:51:34 PM	Completed	Get-MailboxStatistics -Ident...
9/11/2012 5:51:37 PM	9/11/2012 5:51:37 PM	Completed	Get-User -Identity 'DOM2K...
9/11/2012 5:51:47 PM	9/11/2012 5:51:48 PM	Completed	Set-User -HomePhone '111...
9/11/2012 5:51:48 PM	9/11/2012 5:51:48 PM	Completed	Get-Recipient -PropertySet ...
9/11/2012 5:51:48 PM	9/11/2012 5:51:48 PM	Completed	Get-User -Identity 'DOM2K...
9/12/2012 11:18:09 AM	9/12/2012 11:18:10 AM	Completed	Get-Recipient -PropertySet ...

Command:
Get-ExchangeServer| Filter-ExchangeServer -minVersion 14 -maxVersion 14

Output message of command execution:

1 command(s) selected

Command logging started at 9/11/2012 4:46:43 PM

Basics

- Local Shell
  - Exchange 2007
  - Windows PowerShell host
  - Windows PowerShell snap-in, contains exchange cmdlets
  - Custom Scripts
  - Cmdlet is always run on the local Exchange server
- Remote Shell
  - Exchange 2010
  - Powershell.exe remote connection/session, No Snapin loaded
  - Uses Windows Remote Management 2.0
  - Gives access to cmdlets that are assigned based on management role
  - No need for Exchange Management Tools, but recommended

Note : When you open shell on exchange 2010 server two sessions are created local and remote

Edge Role

- What is that ?
- Uses the local Shell only on the Edge Transport server role
- Administered individually

Remote PowerShell

- Connect to an Exchange 2010 from a client with WMF installed and no 2010 management tools
- $userCred = Get-Credential
- $session = New-PSSession -Configurationname Microsoft.Exchange –ConnectionUri http://casservernamefqdn/powershell -Credential $userCred
- Import-PSSession $session
- Remove-PSSession $session
- Set-User kamal-RemotePowerShellEnabled $True

Verbs and Nouns

- New
- Get
- Set
- Add
- Update
- Remove
- Enable
- Disable
- Mount
- Dismount
- Test
- Stop
- Start
- Resume
- Retry

- MailboxDatabase
- SendConnector
- Mailbox
- TransportAgent
- ActiveSyncDevice
- SystemHealth
- ServiceHealth
- JournalRule
- MapiConnectivity
- DistributionGroup
- MailboxDatabaseCopyStatus

cmdlets

- Get-Command
- Get-ExCommad
- Get-Command *mailbox*
- Get-Command –Noun Mailbox
- Get-Command –Verb Restore

Tip: Start-Transcript

Objects

Get-Car

Name	What_I_Wanted
Type:	Classic
Color:	Red
Speed:	100
Mileage:	18

Name	What_I_Have
Type:	Family
Color:	Blue
Speed:	60
Mileage:	35

Name	What_I_Should
Type:	Green
Color:	Green
Speed:	20
Mileage:	55

Name	What_I_Deserve
Type:	Not Affordable
Color:	Red
Speed:	250
Mileage:	8

Change Property

Set-Car “What_I_Have” -Color: Sparkling Bronze Metallic

Methods

Start

Drive

Stop

Lets look at an example

- Get-Service

The Power of TAB

- PowerShell auto completion
  - Auto complete.
    - Try typing **get-a**<TAB>
  - Scroll through parameters or cmdlets.
    - Try typing **Get-Mailbox -<**TAB>

Get-Help

- Use help to find cmdlets and categories
  - Get-help *User*
    Get-help -role *UM*
    Get-help –component *recipient*
- Use help pages to drill into more detail
  - Get-mailbox -? | more
  - Help set-mailbox
  - Get-help get-mailbox –Detailed
  - Get-help set-mailbox –Examples
  - Get-help set-mailbox –Online
  - Get-Help <cmdlet> -Parameter <parameter name>

Alias

- PowerShell Shorthand Notation
  - Aliases are used to shorten common commands in PowerShell.
  - Use get-alias to see all aliases
  - Create your own alias using new-alias

Alias	Cmdlet
dir	get-childitem
cd	set-location
rm	remove-item
rmdir	remove-item
copy	copy-item
echo	write-output
del	remove-item
move	move-item

Parameters

- Provide information to the cmdlet
- Control how the cmdlet performs its task
- Verb-Noun -ParameterName <ParameterValue>
- Types
  - Positional (Identity)
  - Named (Specify the Parameter)
  - Boolean($true, $false)
  - Switch(confirm)
  - Common(Verbose, Debug, ErrorAction)

Syntax

- Verb-Noun -ParameterName <ParameterValue>
- hyphen indicates a parameter
- Space in Parameter Value:Double quotation marks (")
- Single Quote vs Double Quote
  - "$Server Example“
  - ‘$Server Example’
- Escape Character
  - “Gold is `$1600“

Exploring Parameters

- Explore the properties of output objects using format-list
  - Get-Mailbox TestUser | format-list
  - Get-Mailbox TestUser | fl *quota*
  - Get-Mailbox TestUser | fl *
  - Get-ExchangeServer –Status | fl *
  - Get-Mailbox | FT Name,Database
- Explore the property types of output objects using get-member
  - Get-storagegroup TestUser | get-member
- Tab it Set-Mailbox-<tab>

Operators

- **=** value on the right side of the equal sign is assigned to the variable on the left side
- **!** logical **NOT** operator . How do I say “Not Equal To”
- **>, >>** send the output of a command to a file
- **{ }** – Expression
- **$** Variable
- +,-,*,%
- -eq, -ne, -Like, -and, -or, -gt, -lt
  http://technet.microsoft.com/en-us/library/bb125229

Tip: Tee-Object

Pipeline

- What is pipe between cmdlets?

Use Cases for Pipeline

- Bulk management is possible using pipelining

  Get-Mailbox |
  Set-Mailbox -param1 arg1 -param2 arg2
- Piping (cmd1 | cmd2) works within same noun

  Get-Mailbox contoso\joe | remove-mailbox
- And certain different nouns

  Get-Mailbox contoso\joe | Test-MapiConnectivity

What are we talking about - Demo

Process Data

- **Get-Car** Where **Color** is not **Red**

$_

- **Get-Car** Where **Color** is not **Red**

Name	What_I_Wanted
Type:	Classic
Color:	Red
Speed:	100
Mileage:	18

Name	What_I_Have
Type:	Family
Color:	Blue
Speed:	60
Mileage:	35

Name	What_I_Should
Type:	Green
Color:	Green
Speed:	20
Mileage:	55

Name	What_I_Deserve
Type:	Not Affordable
Color:	Red
Speed:	250
Mileage:	8

$_.Color
$_.Speed
$_.Type

$_.Color
$_.Speed
$_.Type

Lets Confuse you

- Get-Mailbox | Where-Object {$_.Name –like “*admin*”}
- Get-Mailbox | ? {$_.Name –like “*admin*”} | Set-Mailbox -ProhibitSendReceiveQuota 10GB
- Get-Mailbox | ? {$_.Name –like “*admin*”} | Select-Object Name, ProhibitSendReceiveQuota | Export-Csv -Path c:\Export.csv

Pipelining to Pipe Data between Dissimilar Nouns

- Use the data from one cmdlet with another cmdlet
- Haven't been optimized to pass objects directly

```
Get-Mailbox | Set-Mailbox
```

VS

```
Get-Mailbox | New-InboxRule
```

```
New-InboxRule -Name "Mark as Read" -Mailbox TEST
```

So .. How do we do it ?

- We Know about $_

Get-Mailbox | **ForEach-Object** {Write-Host $_.Name}

Get-Mailbox | **ForEach** { New-InboxRule -Name "Mark as Read" -Mailbox **$_.Name** -From john@contoso.com -MarkAsRead $True}

WhatIf, Confirm, and ValidateOnly Switches

- Whatif
  - Objects that would be affected by running the command and what changes would be made to those objects
- Confirm
  - Stop processing before any changes are made
- ValidateOnly
  - Evaluate all the conditions and requirements that are needed to perform the operation before you apply any changes

Multi Valued

- @{Add="chris@contoso.com"}
- @{Remove="david@contoso.com"}

Variables

- $
- $CurrentDate = Get-Date
- $CurrentDate | Get-Member

Filtering

- Built in –filter
  - Get-mailbox –filter {alias–like “ka*”}
- Wildcard
  - Get-mailbox admin*
  - Get-ExchangeServer *North*
  - Get-SendConnector *.test.com
- Where-object (alias where)
  - Get-mailbox | where {$_.Alias –like “*admin*”}
  - Get-TransportServer | where { $_.MessageTrackingLogEnabled –eq $false }

Working with Command Output

- Format-list (FL)
  - Get-Mailbox | FL
  - Get-Mailbox | FL *
  - Get-Mailbox | FL *Quota*
- Format-table (FT)
  - Get-Mailbox | FT
  - Get-Mailbox | FT *
  - Get-Mailbox | FT Name, Alias,Database
- Sort-object (sort)
  - Get-mailboxstatistics | sort –property itemcount –desc
- Group-object (group)
  - Get-mailbox | group –property Database

Import

- Get-Content
  - Un Structured data

- Import-Csv
  - Structured data
  - First row = Properties Names
  - All other rows are data
  - Import creates the objects to work with other PowerShell cmdlets

Examples

- To find the number of mailboxes per database:
  - Get-mailbox | group –property database
- To find all users who are nearing or over their quota limit:
  - Get-mailboxstatistics | where {$_.storagelimitstatus –ne ‘BelowLimit’}
- To find all folders with more than 5000 items:
  - Get-mailboxfolderstatistics | where {$_.itemsinfolder –gt 5000}
- To find queues in the retry state:
  - Get-queue | where {$_.status –eg ‘Retry”}
- Get Mailbox Database Size
  - Get-MailboxDatabase -Status | select ServerName,Name,DatabaseSize

Useful Links

- http://technet.microsoft.com/en-us/scriptcenter/powershell.aspx
- http://technet.microsoft.com/en-us/scriptcenter/default
- http://gallery.technet.microsoft.com/ScriptCenter/
- http://www.microsoft.com/en-us/download/details.aspx?id=7097
- http://www.microsoft.com/en-us/download/details.aspx?id=6268
- http://technet.microsoft.com/en-us/library/hh848797
- http://blogs.technet.com/b/heyscriptingguy/