



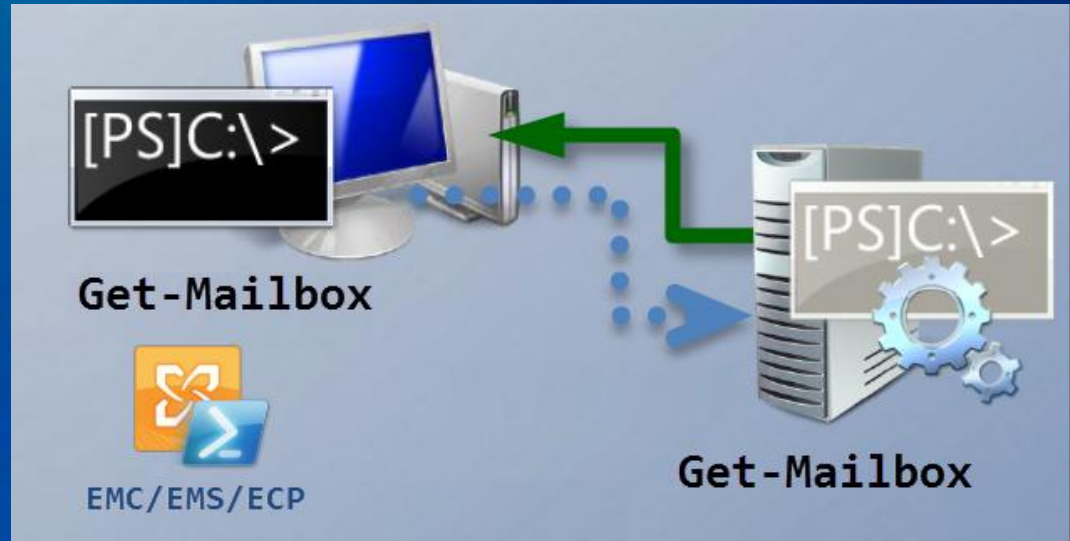
Enterprise Exchange Management with PowerShell



Mike Pfeiffer
Systems Instructor
Interface Technical Training

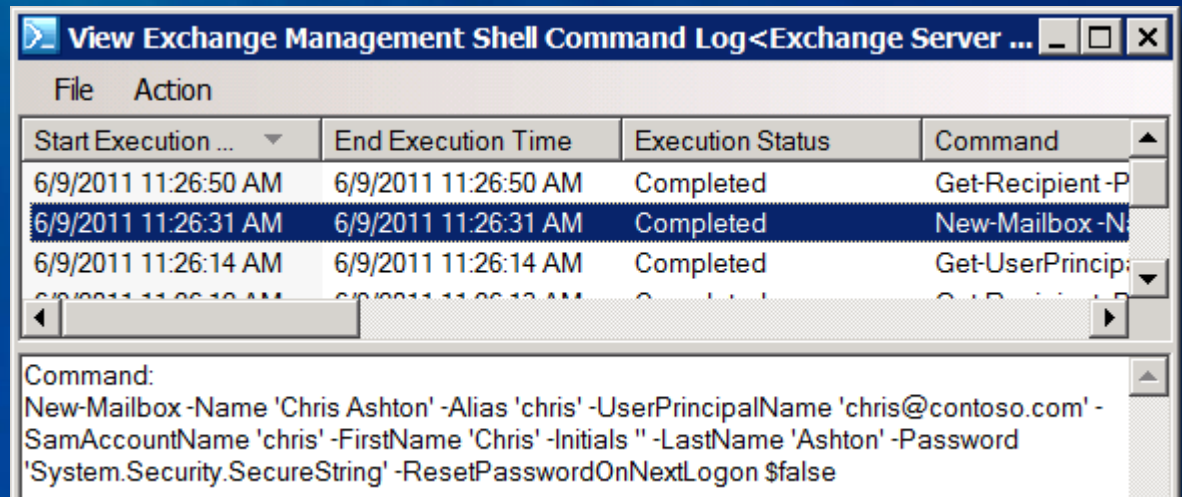
Management Tools Architecture

- PowerShell v2 & WinRM
- All Exchange 2010 Management Tools Built on Remote PowerShell
- Can be run with or without tools installed



Command Exposure

- EMC Command Button
- EMS Commands in Wizards
- EMS Command Logging



The screenshot shows a window titled "View Exchange Management Shell Command Log <Exchange Server ...". It contains a table with columns for "Start Execution Time", "End Execution Time", "Execution Status", and "Command". The table lists several commands executed on 6/9/2011 at 11:26 AM, all with a status of "Completed". The command "New-Mailbox -Name 'Chris Ashton' -Alias 'chris' -UserPrincipalName 'chris@contoso.com' -SamAccountName 'chris' -FirstName 'Chris' -Initials '' -LastName 'Ashton' -Password 'System.Security.SecureString' -ResetPasswordOnNextLogon \$false" is highlighted in blue. Below the table, the full command text is displayed in a text area.

Start Execution ...	End Execution Time	Execution Status	Command
6/9/2011 11:26:50 AM	6/9/2011 11:26:50 AM	Completed	Get-Recipient -P
6/9/2011 11:26:31 AM	6/9/2011 11:26:31 AM	Completed	New-Mailbox -N
6/9/2011 11:26:14 AM	6/9/2011 11:26:14 AM	Completed	Get-UserPrincip
6/9/2011 11:26:10 AM	6/9/2011 11:26:10 AM	Completed	Get-UserPrincip

Command:
New-Mailbox -Name 'Chris Ashton' -Alias 'chris' -UserPrincipalName 'chris@contoso.com' -SamAccountName 'chris' -FirstName 'Chris' -Initials '' -LastName 'Ashton' -Password 'System.Security.SecureString' -ResetPasswordOnNextLogon \$false

Remoting

- Entire Toolset Built on Remoting
- Can Configure Manually:
 - On-Premise
 - Office 365

```
Machine: mbx1.contoso.com
[PS] C:\>Get-Mailbox -OrganizationalUnit Sales

Name                Alias                ServerName           ProhibitSe
ndQuota
-----                -
Alexandra Avery     aavery              mbx1                 unlimited
Roberta Burt        rburt               mbx1                 unlimited
Wanda Jordan        wjordan             mbx1                 unlimited
Jaime Carrillo      jcarrillo           mbx1                 unlimited
Alexandra Benjamin abenjamin           mbx1                 unlimited

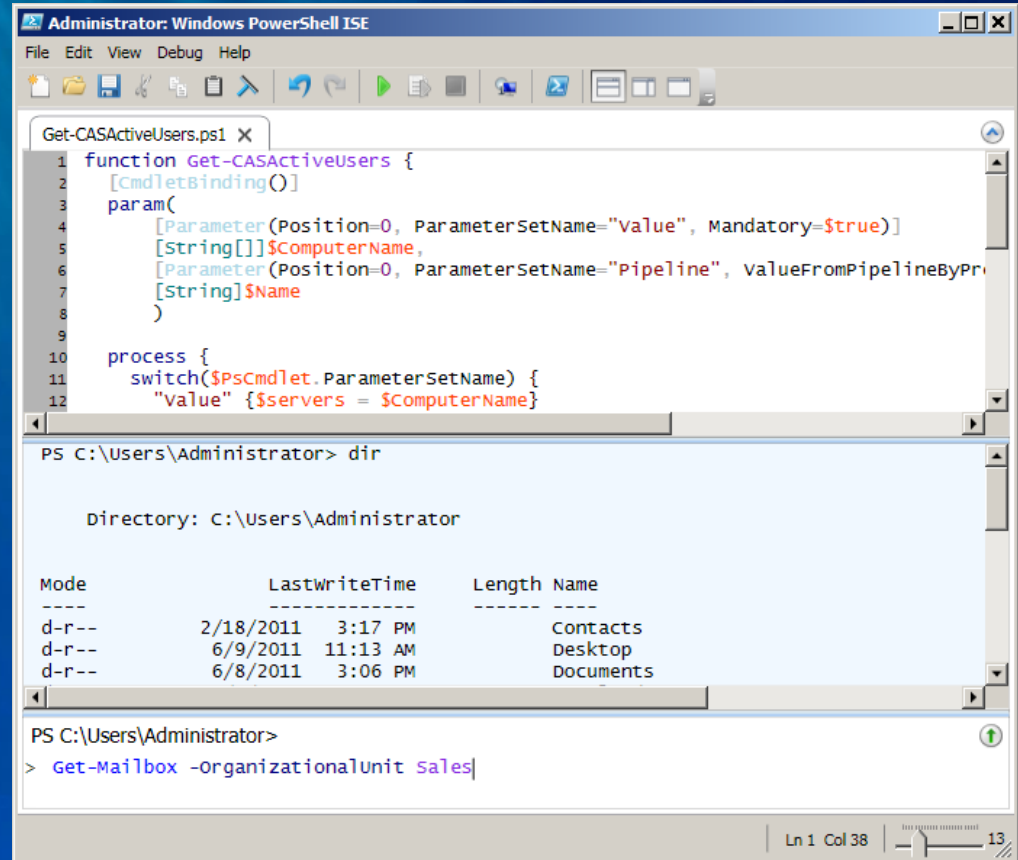
Administrator: Windows PowerShell
PS C:\> Get-Mailbox -OrganizationalUnit Sales

Name                Alias                ServerName           ProhibitSe
ndQuota
-----                -
Alexandra Avery     aavery              mbx1                 unlimited
Roberta Burt        rburt               mbx1                 unlimited
Wanda Jordan        wjordan             mbx1                 unlimited
Jaime Carrillo      jcarrillo           mbx1                 unlimited
Alexandra Benjamin abenjamin           mbx1                 unlimited

PS C:\>
```

Scripting

- Task Automation
- Save as .ps1
- Execution Policy
- Script Editors
 - PowerShell ISE
 - PowerGUI



The screenshot shows the Windows PowerShell ISE interface. The top pane displays a PowerShell script named 'Get-CASActiveUsers.ps1'. The script defines a function 'Get-CASActiveUsers' with a 'value' parameter set. The bottom pane shows the execution of the 'dir' command, listing the contents of the 'C:\Users\Administrator' directory. Below that, the 'Get-Mailbox' command is being typed into the console.

```
function Get-CASActiveUsers {  
    [CmdletBinding()]  
    param(  
        [Parameter(Position=0, ParameterSetName="value", Mandatory=$true)]  
        [String[]]$ComputerName,  
        [Parameter(Position=0, ParameterSetName="Pipeline", valueFromPipelineByPr  
        [String]$Name  
    )  
  
    process {  
        switch($PsCmdlet.ParameterSetName) {  
            "value" {$servers = $ComputerName}        }  
    }  
}
```

PS C:\Users\Administrator> dir

Directory: C:\Users\Administrator

Mode	LastWriteTime	Length	Name
d-r--	2/18/2011 3:17 PM		Contacts
d-r--	6/9/2011 11:13 AM		Desktop
d-r--	6/8/2011 3:06 PM		Documents

PS C:\Users\Administrator>
> Get-Mailbox -organizationalUnit sales|

Bulk Operations

- Some cmdlets designed to work together
- Make use of the PowerShell Pipeline
- Understand Looping Constructs

```
Machine: mbx1.contoso.com
[PS] C:\>Get-Mailbox -OrganizationalUnit Sales |
>> Set-Mailbox -Office Sales
>>
[PS] C:\>

Machine: mbx1.contoso.com
[PS] C:\>Get-Mailbox -OrganizationalUnit Sales |
>> Add-DistributionGroupMember Sales
>>
[PS] C:\>_
```

Reporting

- Export to Text
- Export to CSV
- Export to HTML
- Send as E-Mails

The image shows a screenshot of an Excel spreadsheet and a web browser window. The Excel spreadsheet displays a table with columns: Name, Database, ServerName, and Alias. The web browser window shows the same data rendered as an HTML table.

Name	Database	ServerName	Alias
Alexandra Avery	Mailbox Database 1158621229	mbx1	aavery
Roberta Burt	Mailbox Database 1158621229	mbx1	rburt
Wanda Jordan	Mailbox Database 1158621229	mbx1	wjordan
Jaime Carrillo	Mailbox Database 1158621229	mbx1	jcarrillo
Alexandra Benjamin	Mailbox Database 1158621229	mbx1	abenjamin

Script Automation

- Schedule Scripts with Task Scheduler
- Use Scripting Agent

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration version="1.0">
  <Feature Name="MailboxProvisioning" Cmdlets="New-Mailbox">
    <ApiCall Name="OnComplete">
      if($succeeded) {
        $mailbox = $provisioningHandler.UserSpecifiedParameters["Name"]
        Set-Mailbox $mailbox -SingleItemRecoveryEnabled $true
      }
    </ApiCall>
  </Feature>
</Configuration>
```


Auditing

- Administrator Audit Logging
- Mailbox Audit Logging

```
Machine: mbx1.contoso.com
[PS] C:\>Search-AdminAuditLog |
>> select CmdletName,RunDate,Succeeded,Caller,ObjectModified
>>

CmdletName      : Remove-Mailbox
RunDate         : 6/9/2011 2:18:49 PM
Succeeded       : True
Caller          : contoso.com/Users/Administrator
ObjectModified  : contoso.com/Sales/Roberta Burt
```

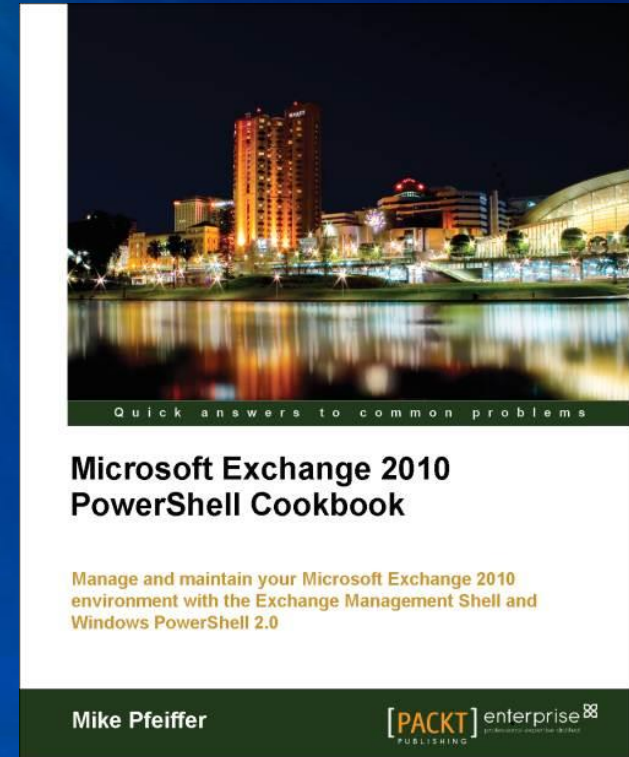
Best Practices & Useful Techniques

- Use a Script Editor
- Use Full Command Names in Scripts
- Use Comments in Code
- Utilize the Help System:
 - Get-Help <cmdlet> -Full
 - Get-Help <cmdlet> -Examples
- Use Get-Command and GetExCommand for Discovery

Exchange 2010 PowerShell Cookbook

- Bulk Operations
- Reporting & Alerting
- Script Automation
- Auditing
- Monitoring & Troubleshooting
- and more...

Pre-Order at: www.packtpub.com



Thank You!

- Blog: www.mikepfeiffer.net
- E-Mail: mike.pfeiffer@interfacett.com
- Twitter: [@mike_pfeiffer](https://twitter.com/mike_pfeiffer)