Exchange

FONSTONE

InfoNet Day 2014
Exchange Server 2013 - Tips & Tricks

Thomas Dehn, Solutions Architect
thomas.dehn@fonstone.com

Pascal Riedo, Solutions Architect
pascal.riedo@fonstone.com

Vorstellung Referenten

Thomas Dehn, Solutions Architect, Fonstone AG

In den vergangenen 16 Jahren arbeitete Thomas Dehn vornehmlich in Enterprise Messaging-Projekten und durfte so bislang nahezu 50‘000 Mailboxen direkt migrieren oder deren Migration begleiten.

Er wird häufig in alle Projektphasen involviert und ist bis heute in Messaging-Projekten bei Konzeptionen, Planungen und deren technischen Umsetzungen aktiv.

Pascal Riedo, Solutions Architect, Fonstone AG

Seine Kerntätigkeiten sind die Beratung, Planung, Konzeption und die technische Umsetzung primär im Rahmen von Enterprise Messaging Projekten.

Er hat erfolgreich für nationale, internationale Konzerne wie auch für Schweizer KMU’s gearbeitet und wird oftmals in alle Projektphasen involviert. Aufgrund seiner Aktivitäten in verschiedensten Kundenprojekten hat er umfangreiche Erfahrungen über Projektabläufe, Prozesse bis hin zum Betrieb der erstellten Plattform.

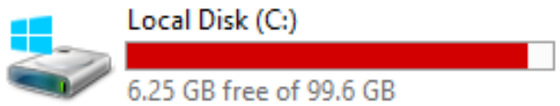
FONSTONE

Agenda

- Logfile CleanUp
- Migrate Receive Connectors
- Export Exchange Settings
- Desaster Recovery
- Server Patching and Maintenance Mode
- Notes on Service Pack 1 … and beyond!

Logfile CleanUp (Purge)

Logfile CleanUp #1/6



Root cause

Exchange 2013 servers do accumulate a lot of Exchange and IIS log files over time. Much more than in Exchange 2010. Even when the server is doing nothing, it is recording the health of the server and writing that down to the logs. To avoid problems some administrators configure Exchange and IIS to store logs on a different disk, while others just wait for free disk space warnings and manually remove old logs.

Additional solution approach

Simple PowerShell script for automating the cleanup of IIS log files as well as Exchange Logging files, based on a defined timeframe you decide to keep the logfiles.

The following presented PowerShell script removes log files (*.log) over 35 days old in the IIS logs folder and the same from the Exchange 2013 logging folder. Neither of these folders are cleaned up automatically in Exchange 2013 RTM or SP1 or CU6.

Logfile CleanUp #2/6

Built-In Logfile maintenance

Transport Logs (e.g. Message Tracking, Send/ReceiveProtocolLogs, ConnectivityLogs,…)

⇨ default location = <ExchangeInstallPath>\TransportRoles\Logs*

Transaction Logs (Backup based Logfile Truncation, Circular Logging)

Logfiles requiring special attention

Additional Exchange Logging (e.g. CmdletInfra, Monitoring, Search, Diagnostics,…)

⇨ default location = <ExchangeInstallPath>\Logging*

Internet Information Server (IIS) Logs

⇨ default location = <SystemDrive>\inetpub\logs\LogFiles*

Logfile CleanUp #3/6

Purge Logfiles Function

```
Function CleanLogfiles($E2K13Srv, $TargetFolder, $LogDays)
{
 $TargetServerFolder = \\$E2K13Srv\$TargetFolder
 if (Test-Path $TargetServerFolder) {
     $Now = Get-Date; $Files = @()
     $LastWrite = $Now.AddDays(-$LogDays)
     [array]$Files = Get-ChildItem $TargetServerFolder -Include *.log -Recurse | Where {$_.LastWriteTime -le "$LastWrite"}
     If ($Files.count -gt 0) {
         Write-Host "Removing files '$($Files.Count)' older than '$LastWrite' from Log-Directory: $TargetServerFolder" -Foregroundcolor Green
         Foreach ($File in $Files) {
             Write-Host "Deleting file $File" -ForegroundColor "Red"
             Remove-Item $File -ErrorAction SilentlyContinue | out-null
         }
     } Else {Write-Host "No Files older than '$LastWrite' to remove from Log-Directory: $TargetServerFolder" -Foregroundcolor Yellow}
 } Else {Write-Host "The folder $TargetServerFolder doesn't exist! Check the folder path!" -ForegroundColor "red"}
}
```

Logfile CleanUp #4/6

Gather Server & Log-Directories and call Purge Logfiles Function

```
$LogWindowsDays=35 # Change the number of days here

# Simple Version: Get local logfile locations & expect Exchange Servers are configured identically for remote logfile purging
Import-Module WebAdministration
$PathIISLog = (Get-WebConfigurationProperty "/system.applicationHost/sites/siteDefaults" -name logfile.directory).Value
$PathIISLogUNC = $PathIISLog.Replace("%SystemDrive%",$env:SystemDrive).Replace(":","$")
$PathExchangeLogging = $env:ExchangeInstallPath + "Logging\"
$PathExchangeLoggingUNC = $PathExchangeLogging.Replace(":","$")

# Get Exchange 2013 Servers
$AllE2013Srv = Get-ExchangeServer | Where {$_.IsE15OrLater -eq $true}

# Call CleanLogfiles Function for each desired Loggingfolder
Write-Host "Removing IIS and Exchange Logs, older than: $LogWindowsDays" -ForegroundColor Green
Foreach ($E2013Srv in $AllE2013Srv) {
    CleanLogfiles $E2013Srv.Name $PathIISLogUNC $LogWindowsDays
    CleanLogfiles $E2013Srv.Name $PathExchangeLoggingUNC $LogWindowsDays
}
```

Logfile CleanUp #5/6

Script Output

Select Machine: EX1301.lab.local

```
[PS] C:\sysman\scripts>.\Logs.CleanUp.ps1

Removing files '2' older than '09/13/2014 11:02:11' from Log-Directory: \\EX1301\C$\inetpub\logs\LogFiles\
Deleting file \\EX1301\C$\inetpub\logs\LogFiles\W3SVC1\u_ex140612.log
Deleting file \\EX1301\C$\inetpub\logs\LogFiles\W3SVC2\u_ex140612.log

Removing files '66' older than '09/13/2014 11:02:13' from Log-Directory: \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061209-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061210-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061211-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061212-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061213-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061214-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061215-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061216-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061217-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061218-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061219-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061220-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061221-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061222-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061223-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061300-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061301-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061302-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061303-1.LOG
Deleting file \\EX1301\C$\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Others\Cmdlet\MSExchangeHMWorker.exe_7644_Cmdlet_2014061304-1.LOG
...
```

Logfile CleanUp #6/6

Possible enhancements

1. Implement the Logfile CleanUp Script as daily scheduled maintenance task (for each Exchange Server, locally or remotely)

2. Before removing logfiles copy them to another location e.g. zipped

3. Determine IIS Logfiles Path for each remote Exchanger server individually
4. Determine Exchange Install Path for each remote Exchanger servers individually and determine customized Exchange Logging Path configurations

5. Write a logfile ☺ instead of standard output only

6. Handle dedicated Exchange Server roles seperately if not multirole servers only in place

Exchange

FONSTONE

Migrate Receive Connectors

Migrate Receive Connectors #1/4

Receive Connectors (RC)

Receive Connectors are configured per Exchange Server

Additional Receive Connectors are often created and operated
(e.g. managed SMTP internal delivery or external relay for application servers)

Their configuration should be kept the same on all servers where a same RC is configured

Migrate or just copy RC configuration

Transfer of current RC configuration in a migration scenario or just to an additionally installed Exchange Server is required from time to time and executed manually it is laborious and error-prone.

e.g. because of extensive RemoteIPRanges lists or any non-default configuration detail

Migrate Receive Connectors #2/4

Example Basic Script

```
# Exchange 2013 Mail flow configuration - Create new E2K13 RECEIVE Connectors based on existing one(s) from E2K7
$SourceServer = "E2K7-SRV01"
$E2K13Servers = "E2K13-SRV01,E2K13-SRV03,E2K13-SRV03"

$ReceiveConnectors = Get-ReceiveConnector -Server $SourceServer | where {($_.Name -notlike "Client*") -and ($_.name -notlike "Default*")}
$TargetServers = $E2K13Servers.Split(",") | Get-ExchangeServer

Foreach ($TargetServer in $Targetservers) {
    Foreach ($RC in $ReceiveConnectors) {
    # Copy basic RC settings:
    New-ReceiveConnector -Server $TargetServer.Name -Name $RC.name -Bindings $RC.Bindings -RemoteIPRanges $RC.RemoteIPRanges
        -AuthMechanism $($RC.AuthMechanism) -PermissionGroups Anonymous -MaxMessageSize $RC.MaxMessageSize -TransportRole:FrontendTransport
    }
}
```

Migrate Receive Connectors #3/4

Possible enhancements

1. Restore Anonymous Relay if PermissionGroup «custom» was used

2. Export/Import Setting via XML ⇨ see next Tip

3. Include additional Receive Connector Settings

   -AuthMechanism | -Banner | **-Bindings** | -ChunkingEnabled | -Comment | -ConnectionTimeout | -ConnectionInactivityTimeout | -DefaultDomain | -DeliveryStatusNotificationEnabled | -DomainSecureEnabled | -EightBitMimeEnabled | -EnableAuthGSSAPI | -ExtendedProtectionPolicy | -ExtendedProtectionTlsTerminatedAtProxy | -LongAddressesEnabled | -Enabled | -MessageRateLimit | -MaxInboundConnection | -MaxInboundConnectionPerSource | -MaxInboundConnectionPercentagePerSource | -MaxHeaderSize | -MaxHopCount | -MaxLocalHopCount | -MaxLogonFailures | **-MaxMessageSize** | -MaxProtocolErrors | -MaxRecipientsPerMessage | **-Name** | -OrarEnabled | **-PermissionGroups** | -PipeliningEnabled | -ProtocolLoggingLevel | -RequireEHLODomain | **-RemoteIPRanges** | -RequireTLS | -SizeEnabled | -TarpitInterval | **-TransportRole**

Migrate Receive Connectors #4/4

Example Basic (+Relay) Script

```
# Exchange 2013 Mail flow configuration - Create new E2K13 RECEIVE Connectors based on existing one(s) from E2K7
$SourceServer = "E2K7-SRV01"
$E2K13Servers = "E2K13-SRV01,E2K13-SRV03,E2K13-SRV03«

$ReceiveConnectors = Get-ReceiveConnector -Server $SourceServer | where {($_.Name -notlike "Client*") -and ($_.name -notlike "Default*")}
$TargetServers = $E2K13Servers.Split(",") | Get-ExchangeServer

Foreach ($TargetServer in $Targetservers) {
    Foreach ($RC in $ReceiveConnectors) {
    # Copy basic RC settings:
    New-ReceiveConnector -Server $TargetServer.Name -Name $RC.name -Bindings $RC.Bindings -RemoteIPRanges $RC.RemoteIPRanges
        -AuthMechanism $($RC.AuthMechanism) -PermissionGroups Anonymous -MaxMessageSize $RC.MaxMessageSize -TransportRole:FrontendTransport

    # Allow Anonymous Relay if PermissionGroup "custom" was used:
    If ($RC.PermissionGroups -match "Custom"){
        $rcid = $TargetServer.Name + "\" + $RC.name
        Get-ReceiveConnector -Identity "$RCID" | Add-ADPermission -User "NT AUTHORITY\ANONYMOUS LOGON"
                -ExtendedRights "Ms-Exch-SMTP-Accept-Any-Recipient«
        }

    }
}
```

Restore Anonymous Relay if PermissionGroup «custom» was used

FONSTONE

Export Exchange Settings

How to store Exchange Settings #1/6

The Quest

If you are applying changes to Exchange settings or want to store for other reason current Exchange configuration information (such as user email addresses, virtual directory or receive connectors configurations). You can easily store configuration data that you might need for comparison and retrieve easily or even restore it afterwards.

Overcome Challenge Multivalued Attributes

Often **Export-Csv** is used to export configurations data e.g. because of its easy-interpretable format (Notepad, Excel). In this case you additionally need to overcome the challenge to export multivalued attribute content.

Tip: Export and store configuration data in XML files.

How to store Exchange Settings #2/6

Export-Csv; Import-Csv

If you don't specifically address multivalued attributes then the content of such attributes is not available in your file export but gets typically replaced by:

```
Microsoft.Exchange.Data.MultiValuedProperty`1[System.Globalization.CultureInfo]
Microsoft.Exchange.Data.MultiValuedProperty`1[System.String]
Microsoft.Exchange.Data.MultiValuedProperty`1[System.Guid]
Microsoft.Exchange.Data.Directory.ADMultiValuedProperty`1[Microsoft.Exchange.Data.Directory.MailboxProvisioningConstraint]
…
```

How to store Exchange Settings #3/6

Export-Clixml; Import-Clixml

Export-Clixml allows including **ALL** properties without any additional data handling.

Import-Clixml allows us to retrieve our original exportet configuration (Get-Mailbox, Get-ReceiveConnector, …) we piped to Export-Clixml, including **ALL** the properties that have been available and included at that time.

Therefore we can use the same approach to examine what the configuration has been as we would query current values from Exchange Servers now.

With the difference of loading the data with Import-Clixml first.

How to store Exchange Settings #4/6

Export-Csv; Import-Csv

Select Machine: EX1301.lab.local

```
[PS] C:\temp>Get-Mailbox pascal.riedo | FL SamAccountName,EmailAddresses,PrimarySmtpAddress


SamAccountName     : pascal.riedo
EmailAddresses     : {smtp:rip@lab.local, SMTP:pascal.riedo@lab.local}
PrimarySmtpAddress : pascal.riedo@lab.local

[PS] C:\temp>Get-Mailbox pascal.riedo | Export-Csv .\pascal.riedo.csv

[PS] C:\temp>Import-Csv .\pascal.riedo.csv | FL SamAccountName,EmailAddresses,PrimarySmtpAddress


SamAccountName     : pascal.riedo
EmailAddresses     : Microsoft.Exchange.Data.ProxyAddressCollection
PrimarySmtpAddress : pascal.riedo@lab.local

[PS] C:\temp>
```

How to store Exchange Settings #5/6

Export-Clixml; Import-Clixml

Select Machine: EX1301.lab.local

```
[PS] C:\temp>Get-Mailbox pascal.riedo | FL SamAccountName,EmailAddresses,PrimarySmtpAddress


SamAccountName     : pascal.riedo
EmailAddresses     : {smtp:rip@lab.local, SMTP:pascal.riedo@lab.local}
PrimarySmtpAddress : pascal.riedo@lab.local

[PS] C:\temp>Get-Mailbox pascal.riedo | Export-Clixml .\pascal.riedo.xml

[PS] C:\temp>Import-Clixml .\pascal.riedo.xml | FL SamAccountName,EmailAddresses,PrimarySmtpAddress


SamAccountName     : pascal.riedo
EmailAddresses     : {smtp:rip@lab.local, SMTP:pascal.riedo@lab.local}
PrimarySmtpAddress : pascal.riedo@lab.local

[PS] C:\temp>
```

How to store Exchange Settings #6/6

Additional Examples

```
Get-Mailbox john.doe | Export-CliXml .\john.doe.xml -Encoding Unicode
Get-Mailbox -Resultsize Unlimited | Export-CliXml .\Mbx.xml -Encoding Unicode

Get-ReceiveConnector EX1301\ReceiveInternal-ForRelay | Export-CliXml .\RC_EX1301.xml
Get-ReceiveConnector | Export-CliXml .\RC_All.xml -Encoding Unicode

Get-OwaVirtualDirectory | Export-Clixml .\OWAVirtualDirectories.xml -Encoding Unicode
Get-OutlookAnywhere | Export-Clixml .\OutlookAnywhere.xml -Encoding Unicode

Get-User -ResultSize Unlimited | Export-CliXml .\Usr.xml -Encoding Unicode
```

Note

We use this method to analyze & compare undocumented customer Exchange configurations in order to visualize and even document customizations

How to store Exchange Settings #more tips

Tip: expand multivalued properties

Select examples for expanding multivalued properties

```
Get-Mailbox john.doe | Select EmailAddresses
Get-Mailbox john.doe | Select -ExpandProperty EmailAddresses

Get-Mailbox john.doe | Select Name,
@{Name='EmailAddresses';Expression={[string]::join(";", ($_.EmailAddresses))}}
```

Tip: add/remove multivalued properties

```
Set-Mailbox john.doe -EmailAddresses @{Add='newaddress@lab.local'}
Set-Mailbox john.doe -EmailAddresses @{Remove='newaddress@lab.local'}
```

Exchange

FONSTONE

Desaster Recovery

Re-installing any lost DAG member #1/6

Ever lost a DAG member during CU installation?

- … hopefully not! ☺
- Installation procedure may abort. Even today!
- Node is in an indefinite state then
- Setup may not be capable to resume under any (obvious) circumstances
- You need to decide what to do next!

Things you should NOT consider

- Install the Server freshly with a new name and join it to the DAG
- Use *ADSIEdit* to remove the server and see what happens next
- Turn the machine off and enjoy your weekend

Re-installing any lost DAG member #2/6

But what to do then?

- Remember the TechNet article to recover any server http://tinyurl.com/recoverserver
- … but it's not as straight forward as you may expect!
- It's only a VERY generic approach that needs a few more steps to complete

A few notes for subsequent steps

- DAC mode should always be enabled for DAG to have all needed cmdlets for any DR scenario
- Procedure has been engineered during a real DR scenario at night in a 3 node DAG environment
- Proofed during several DR tests
- Keep your environment documented with at least a minimum set of configuration details!

Re-installing any lost DAG member #3/6

Preparation

1. Save all needed local files on affected server
2. Save/Export needed Exchange certificates (with any private key)
3. Note exact NIC config
4. Note current PageFile size: (typically physical RAM + 10 MB)
5. Note virtual directory settings (external- and internal URLs, authentication)
6. Note activation priority of all databases within DAG
7. Reset computer account in AD

TIP: Use Export-CliXML to capture relevant configuration data

Re-installing any lost DAG member #4/6

Re-Installation

1. Re-install OS. **Important:** use same version as before!
2. Provide same name and IP address(es) for machine and rejoin to domain
3. Patch OS to latest patch level
4. Import previously exported certificates
5. Install E2013 pre-requisites (UM API, Office FilterPack + SP)
6. Adjust PageFile size accordingly
7. Recover DB drives (take online) and ensure drive letters are correct
8. E2013 Installation with binaries from NEW CU!
   `setup /m:recoverserver /iacceptexchangeserverlicenseterms`
   Note: `/mapi` virtual directory will **not** be recreated! See http://support.microsoft.com/kb/2931223
9. Installing all needed UM Language Packs
   `setup /AddUmLanguagePack:de-DE /SourceDir:c:\Installation\E2013_SP1_files /iacceptexchangeserverlicenseterms`

Re-installing any lost DAG member #5/6

Restore potentially lost information

1. Recover virtual directories and SCP

```
$Srv = Get-ExchangeServer <lost node name>
# Virtual Directories
Set-ActiveSyncVirtualDirectory –Server $Srv -InternalURL "https://vip.company.com/Microsoft-Server-ActiveSync"
Set-EcpVirtualDirectory –Server $Srv -InternalURL "https://vip.company.com/ecp"
Set-OabVirtualDirectory–Server $Srv -InternalURL "https://vip.company.com/OAB"
Set-OwaVirtualDirectory–Server $Srv -InternalURL "https://vip.company.com/owa"
Set-WebServicesVirtualDirectory–Server $Srv –InternalURL "https://vip.company.com/EWS/Exchange.asmx"
# SCP
$Srv | Get-ClientAccessServer | select *uri
Set-ClientAccessServer -$Srv –AutoDiscoverServiceInternalUri
        "https://vip.company.com/Autodiscover/Autodiscover.xml"
```

2. Adjust FIPS proxy settings if needed (system proxy with **netsh**)
3. Add node to DAG

```
Add-DatabaseAvailabilityGroupServer -Identity <DAG name> -MailboxServer $Srv.Name
```

Re-installing any lost DAG member #6/6

Restore potentially lost information (Cont.)

1. Add DB replicas to server
```
Add-MailboxDatabaseCopy -Identity <DB01> -MailboxServer $Srv.Name
# PAUSE and check if OK for 1st DB!
Add-MailboxDatabaseCopy -Identity <DB02> -MailboxServer $Srv.Name
# Repeat for all involved DBs
```
2. Eventually reseed failed DBs and wait until everything is healthy again
3. Verify activation priority for databases is still the same as before.
4. Verify the OWA default domain is still the same one the OWA virtual directory
5. Install backup software and other 3rd party software if needed and ensure everything works as before
6. Test if everything is fine again!
7. Don't wonder if Exchange version number seems the old one.
   It IS the correct version, as we used the bits for the re-installation!

Exchange

FONSTONE

Server Patching and Maintenance Mode

Maintenance Mode #1/5

What is Maintenance Mode?

- Prevent Exchange from acting as a CAS and/or MBX server
- Empty queues, take server components offline and pause cluster node

When to take your Server into Maintenance Mode

- patching the OS
- installing Cumulative Updates for Exchange 2013
- maintaining hardware, virtual environments, etc.

How to take your Server into Maintenance Mode

- Use a small scriptlet or a set of cmdlets.
- Don't use any script from `$exscripts` folder! It's not suitable for server components
- Eventually take Server in maintenance on load balancers as well

Maintenance Mode #2/5

- Steps to put Server into Maintenance Mode

  http://tinyurl.com/E2013Maint

1. Drain active mail queues on the mailbox server

```
Set-ServerComponentState <ServerName> -Component HubTransport -State Draining -Requester Maintenance
```

2. To help transport services immediately pick the state change run:

```
Restart-Service MSExchangeTransport
Restart-Service MSExchangeFrontEndTransport # if Server has CAS role installed
```

3. To redirect messages pending delivery in the local queues to another Mailbox server run:

```
Redirect-Message -Server <ServerName> -Target <MailboxServerFQDN>
```

4. To prevents the node from being and becoming the PAM, pause the cluster node by running

```
Suspend-ClusterNode <ServerName>
```

5. To move all active databases currently hosted on the DAG member to other DAG members, run

```
Set-MailboxServer <ServerName> -DatabaseCopyActivationDisabledAndMoveNow $True
```

Maintenance Mode #3/5

6. To prevent the server from hosting active database copies, run

```
Set-MailboxServer <ServerName> -DatabaseCopyAutoActivationPolicy Blocked
```

7. To put the server in maintenance mode run:

```
Set-ServerComponentState <ServerName> -Component ServerWideOffline -State Inactive
        -Requester Maintenance
```

- Steps to take Server out of Maintenance Mode

```
Set-ServerComponentState <ServerName> -Component ServerWideOffline -State Active
        -Requester Maintenance
Resume-ClusterNode <ServerName>
Set-MailboxServer <ServerName> -DatabaseCopyActivationDisabledAndMoveNow $False
Set-MailboxServer <ServerName> -DatabaseCopyAutoActivationPolicy <as it was before>
Set-ServerComponentState <ServerName> -Component HubTransport -State Active
        -Requester Maintenance
Restart-Service MSExchangeTransport
Restart-Service MSExchangeFrontEndTransport # if Server has CAS role installed
```

Maintenance Mode #4/5

Approach to automate e.g. Server OS patching

- Ensure your SW deployment solution is capable to execute PowerShell scripts before and after installing new patches
- Key is to ensure that Server is *entirely* in maintenance mode before patching
- … and back online again, once patching has been done!

<u>Script1: Start-ServerMaintenance</u>

1. Takes Server into maintenance
2. Verifies that server is ready for being patched

<u>Script2: Stop-Servermaintenance</u>

1. Brings server back online
2. Verifies that everything is up and running again
3. (Redistribute databases once last server in DAG has been patched)

Maintenance Mode #5/5

- Reusable technique for similar requirements
- Example function:

```
function Check-MailboxServerStateUP([string] $server){
    $mstateUP = $true
    $mbx = Get-MailboxServer -Identity
    $server | select DatabaseCopy*
        if ($mbx.DatabaseCopyActivationDisabledAndMoveNow -ne $false){$mstateUP = $false}
        if ($mbx.DatabaseCopyAutoActivationPolicy -ne "Unrestricted") {$mstateUP = $false}
    return $mstateUP
}
```

- Get overall Server state

  $overallstate returns a **$true** only if every tested criteria returns a **$true**

```
$srv = «EXSRV01»
$overallstate = $true
    if (!(Check-ComponentStateUP $srv)) {$overallstate = $false}
    if (!(Check-MailboxServerStateUP $srv)) {$overallstate = $false}
    if (!(Check-ClusterNodeStateUP $srv)) {$overallstate = $false}
    if (!(Check-ExServicesUP $srv)) {$overallstate = $false}
    # ...
$overallstate
```

FONSTONE

Exchange

FONSTONE

Questions

Thomas Dehn, Solutions Architect
thomas.dehn@fonstone.com

Pascal Riedo, Solutions Architect
pascal.riedo@fonstone.com

Exchange

FONSTONE

Notes on Service Pack 1

… and beyond!

New in Exchange Server 2013 SP1

- Support for Windows Server 2012 R2

  Install on Windows Server 2012 R2

  Use Windows Server 2012 R2 writable directory servers

  Leverage Windows Server 2012 R2 DFL/FFL

- MAPI over HTTP

  New communication protocol for Outlook 2013 SP1 and later

  Disabled by default in Exchange 2013 SP1

- EAC Command Logging
- DAGs without CAAPs
- Edge Transport Server

MAPI over HTTP – Basics

- New communication protocol in Exchange 2013 SP1 and Outlook 2013 SP1

  Referred to in product as the Exchange HTTP Transport, and by the internal code name Alchemy

- Modernizes the Outlook/Exchange connection by removing dependency on RPC at transport layer

  Outlook continues to use the same ROP verbs to communicate with Exchange

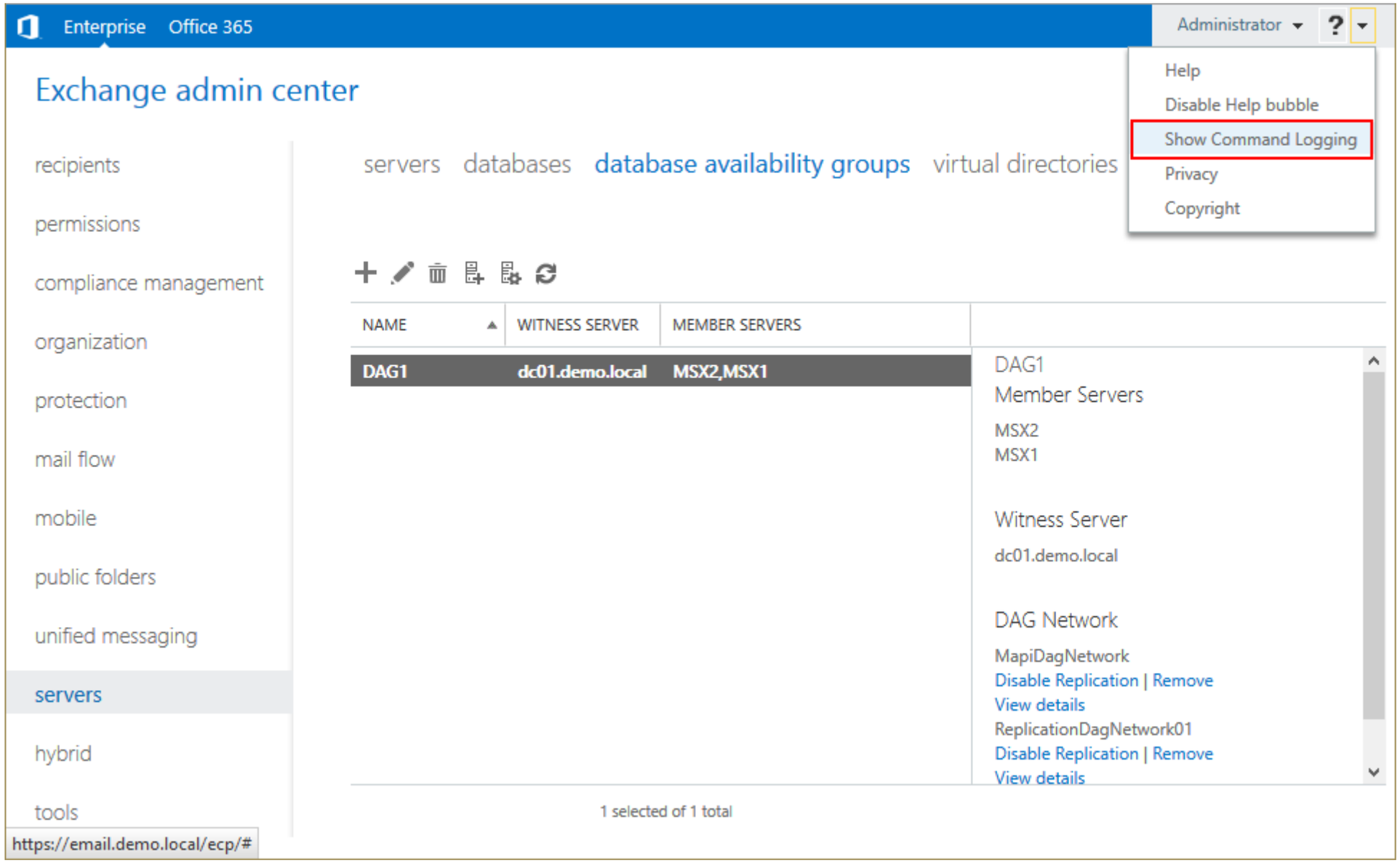
  Outlook creates an HTTP tunnel directly to Exchange

- Outlook connects to the /mapi virdir for mail and directory, not /rpc

  Only mail and directory connect via /mapi

  Still uses /EWS, /OAB, /AutoDiscover, etc., for Web service calls

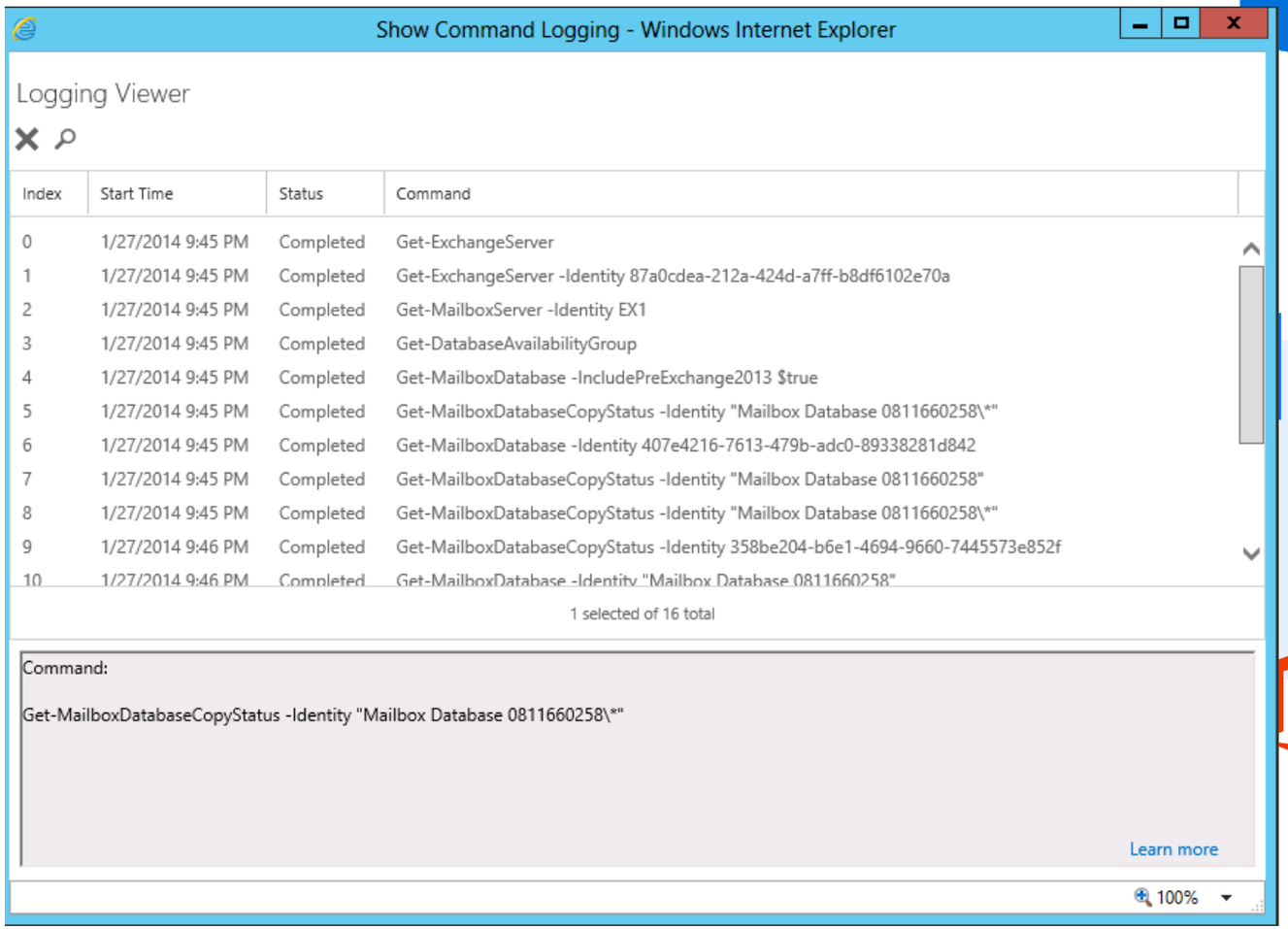
EAC Command Logging – Basics



FONSTONE

EAC Command Logging – Basics

- Must be open to log actions
  Displays up to 500 entries
- Click ✕ to clear log
  Log is cleared when EAC is closed
- Click 🔎 to search log
- Select one or more items
  Multi-select to see cmdlets from multiple items



FONSTONE

Show Command Logging - Windows Internet Explorer

Logging Viewer

Index	Start Time	Status	Command
0	1/27/2014 9:45 PM	Completed	Get-ExchangeServer
1	1/27/2014 9:45 PM	Completed	Get-ExchangeServer -Identity 87a0cdea-212a-424d-a7ff-b8df6102e70a
2	1/27/2014 9:45 PM	Completed	Get-MailboxServer -Identity EX1
3	1/27/2014 9:45 PM	Completed	Get-DatabaseAvailabilityGroup
4	1/27/2014 9:45 PM	Completed	Get-MailboxDatabase -IncludePreExchange2013 $true
5	1/27/2014 9:45 PM	Completed	Get-MailboxDatabaseCopyStatus -Identity "Mailbox Database 0811660258*"
6	1/27/2014 9:45 PM	Completed	Get-MailboxDatabase -Identity 407e4216-7613-479b-adc0-89338281d842
7	1/27/2014 9:45 PM	Completed	Get-MailboxDatabaseCopyStatus -Identity "Mailbox Database 0811660258"
8	1/27/2014 9:45 PM	Completed	Get-MailboxDatabaseCopyStatus -Identity "Mailbox Database 0811660258*"
9	1/27/2014 9:46 PM	Completed	Get-MailboxDatabaseCopyStatus -Identity 358be204-b6e1-4694-9660-7445573e852f
10	1/27/2014 9:46 PM	Completed	Get-MailboxDatabase -Identity "Mailbox Database 0811660258"

1 selected of 16 total

Command:

```
Get-MailboxDatabaseCopyStatus -Identity "Mailbox Database 0811660258\*"
```

Learn more

100%

FONSTONE

Show Command Logging - Windows Internet Explorer

Logging Viewer

Index	Start Time	Status	Command
0	1/27/2014 9:5...	Completed	Get-ExchangeServer
1	1/27/2014 9:5...	Completed	Get-ExchangeServer -Identity 87a0cdea-2...
2	1/27/2014 9:5...	Completed	Get-MailboxServer -Identity EX1
3	1/27/2014 9:5...	Completed	Get-MailboxDatabase -IncludePreExchang...
4	1/27/2014 9:5...	Completed	Get-DatabaseAvailabilityGroup
5	1/27/2014 9:5...	Completed	Get-MailboxDatabase -Identity 407e4216-...
6	1/27/2014 9:5...	Completed	Get-MailboxDatabaseCopyStatus -Identity ...
7	1/27/2014 9:5...	Completed	Get-ClientAccessServer
8	1/27/2014 9:57 ...	Completed	Get-ActiveSyncVirtualDirectory -ADProperties...
9	1/27/2014 9:57 ...	Completed	Get-AutodiscoverVirtualDirectory -ADProperti...
10	1/27/2014 9:57 ...	Completed	Get-EcpVirtualDirectory -ADPropertiesOnly $t...
11	1/27/2014 9:57 ...	Completed	Get-OabVirtualDirectory -ADPropertiesOnly $...
12	1/27/2014 9:57 ...	Completed	Get-OwaVirtualDirectory -ADPropertiesOnly $...

8 selected of 19 total

```
Get-ExchangeServer

Get-ExchangeServer -Identity 87a0cdea-212a-424d-a7ff-b8df6102e70a

Get-MailboxServer -Identity EX1

Get-MailboxDatabase -IncludePreExchange2013 $true

Get-DatabaseAvailabilityGroup

Get-MailboxDatabase -Identity 407e4216-7613-479b-adc0-89338281d842
```

Learn more

100%

FONSTONE

DAGs w/o Cluster Admin Access Points

- Windows Server 2008 R2 and Windows Server 2012 DAGs require at least one IP address on MAPI network

  DAGs require more than one IP address when the MAPI network is extended across subnets

- Windows Server 2012 R2 introduces clusters that can operate without an administrative access point

  No IP Address resource
  No Network Name resource
  No Cluster Name Object (CNO)
  No DNS registration for cluster
  No Failover Cluster Manager access

Overview

A failover cluster is a set of independent computers that work together to increase the availability of server roles. The clustered servers (called nodes) are connected by physical cables and by software. If one of the nodes fails, another node

Nam

Connection Error

Cannot connect without an administrative access point

You cannot use Failover Cluster Manager to manage a cluster that was created without an administrative access point.

You must use Windows PowerShell to manage the cluster.

OK

Management

To begin to use failover clustering, first validate your hardware configuration, and then create a cluster. After these steps are complete, you can manage the cluster. Managing a cluster can include copying roles to it from a cluster running Windows Server 2012 R2, Windows Server 2012, or Windows Server 2008 R2.

- Validate Configuration...
- Create Cluster...

Create Cluste
Connect to C
View
Refresh
Properties
Help

Administrator: Windows PowerShell

```
PS C:\Users\administrator.E15DEMOS> Get-Cluster -Name ex4 | fl


Name : D2


PS C:\Users\administrator.E15DEMOS> Get-ClusterResource -Cluster ex4 | fl


Name          : File Share Witness (\\ex3.e15demos.com\D2.e15demos.com)
State         : Online
OwnerGroup    : Cluster Group
ResourceType  : File Share Witness


PS C:\Users\administrator.E15DEMOS> Get-ClusterResource -Cluster ex5 | fl


Name          : File Share Witness (\\ex3.e15demos.com\D2.e15demos.com)
State         : Online
OwnerGroup    : Cluster Group
ResourceType  : File Share Witness


PS C:\Users\administrator.E15DEMOS> Get-Cluster -Name ex5 | fl


Name : D2


PS C:\Users\administrator.E15DEMOS> _
```

http://aka.ms/PowerShell_WFC

Edge Transport Server – Basics

- Enables customers to use a perimeter network Exchange 2013 server to handle all of the organization's Internet-facing email

  Requires minimum 4GB memory

  Easy to set up

- Designed for minimal attack surface

  No GUI / No ECP – this means IIS not used, which reduces attack surface

  → NO PowerShell connect to EDGE Box

Exchange

FONSTONE

Questions

Thomas Dehn, Solutions Architect
thomas.dehn@fonstone.com

Pascal Riedo, Solutions Architect
pascal.riedo@fonstone.com