PowerShell for Lync Server 2010

Microsoft Lync Server 2010 ships with over 500 Lync-specific PowerShell cmdlets which provide complete management over a Lync deployment. This cheat sheet will help you get started, accomplish frequent management tasks, and provide tips for particular 'gotchas'.

Getting Started

The PowerShell cmdlets for Lync Server are accessible through the **Lync Server Management Shell**. It is installed with "Install Topology Builder" step in the Lync Server Deployment Wizard.

To start it locally:

- **Start** Menu | **All Programs | Microsoft Lync Server 2010 | Lync Server Management Shell**.

Permissions

Lync Server 2010 uses role-based access control (RBAC) to grant or deny administrative privileges. Each Lync cmdlet requires the AD user running the cmdlet to have at least one of the RBAC role(s) assigned to that cmdlet.

There are 11 pre-define Lync Server RBAC roles. For more details see http://technet.microsoft.com/en-us/library/gg425917.aspx.

To see a list of RBAC roles assigned to a user:

```
Get-CsAdminRoleAssignment SamAccountName
```

(**Note:** you must use the SamAccountName when specifying the user Identity for this cmdlet).

To see a list of RBAC roles can run the cmdlet "Cs-CsAdUser":

```
Get-CsAdminRole | Where-Object {$_.Cmdlets -match 'Get-CsADUser'}
```

To grant a Lync RBAC role to a user:

Add the user AD account to the AD security group associated with the Lync RBAC role.

Note: RBAC restrictions in the Lync Server Management Shell only apply remotely. A user running cmdlets in the local Lync Management shell session (e.g. on the server) are not subject to RBAC restrictions.

Getting Help

Get basic help for a cmdlet:

```
Get-Help <cmdlet name>
```

Get parameter descriptions and examples for a cmdlet:

```
Get-Help <cmdlet name> -Full
```

View and search a Lync server cmdlet CHM file:

Lync_ITPro.CHM

(http://www.microsoft.com/downloads/en/details.aspx?FamilyID=9720c3f1-ddd4-426b-b98a-f1205561ce00)

Get a list of all Lync Server commands:

```
Get-Command -Module Lync
```

Do a wildcard search for a cmdlet:

```
Get-Command *CsUser* -CommandType Cmdlet
```

Get all the properties and methods for Get-CsUser:

```
Get-CsUser | Get-Member
```

Get all the properties and values of Lync "User01":

```
Get-CsUser <sip:User01@example.com> | Format-List *
```

For a list of all Lync cmdlets organized by category see http://technet.microsoft.com/en-us/library/gg398867.aspx

Remoting

Most Lync Server PowerShell cmdlets leverage the remote management capabilities of Microsoft PowerShell Version 2.0 to read or write from Lync system objects stored in either the Lync Central Management Store (CMS) or AD. As such, they can be run remotely on any computer where the Lync Server Management is installed.

A small number of Lync cmdlets are specific to a Lync server and the **"-ComputerName"** parameter is needed to identify the server. For example, to return detailed information about the Lync Server 2010 components that runs as Windows services on "Server01" use:

```
Get-CsWindowsService -ComputerName Server01
```

Another small set of cmdlets, such as **Get-CsCertificate** and **Test-CsCertificateConfiguration**, need to be run locally on the Lync computer from which they get or set data from.

Working with Users

Getting User Information

The difference between **Get-CsUser** and **Get-CsAdUser**:

Both cmdlets return information about Lync users. Use Get-CsUser unless you require access to the generic AD attributes, or if the user has not been enabled for Lync Server.

Show all the properties returned by Get-CsUser:

```
Get-CsUser | Get-Member
```

Show the number of Lync enabled users:

```
@(Get-CsUser).Count
```

Get all AD users who are not, and who have never been, enabled for Lync or OCS:

```
Get-CsAdUser -Filter {Enabled -eq $null} | Format-Table -Wrap
```

Get users who are disabled for Lync or OCS:

```
Get-CsAdUser -Filter {Enabled -eq $False}
```

Get all users who have been enabled for Lync or OCS but are not currently assigned to a registrar pool.

```
Get-CsUser -UnassignedUser
```

Get all Lync Users in the Sales Department:

```
Get-CsUser -LDAPFilter 'Department=Sales'
```

Find all Lync users whose name starts with 'Bob':

```
Get-CsUser Bob*
```

Get all the SIP addresses in your Lync deployment:

```
Get-CsUser | Select-Object SipAddress
```

Get several commonly used Lync settings for all Lync users (formatted in a table):

```
Get-CsUser | Format-Table -Wrap -Property DisplayName, SipAddress,
RegistrarPool, VoicePolicy
```

Get all Lync users enabled for Lync Federation:

```
Get-CsUser | Where-Object {$_.EnabledForFederation} | Format-Table `
DisplayName, SipAddress
```

Get the Voice Policy assigned to Lync "User01":

```
(Get-CsUser User01@example.com).VoicePolicy
```

Get all Lync users with the 'Long Distance' Voice Policy:

```
Get-CsUser | Where-Object {$_.VoicePolicy -match "Long Distance"} |
Select-Object DisplayName
```

Get the Primary and Backup pools for "User01":

```
Get-CsUserPoolInfo User01@example.com
```

Get a list of users homed on OCS 2007 R2 or OCS 2007:

```
Get-CsUser -OnOfficeCommunicationServer
```

Get a list of all the user policies assigned to Lync "User01":

```
Get-CsUser user01@example.com | Format-List -Property *Policy*
```

(**Note:** Get-CsUser only shows the User policies that are applied. It does not show whether a **Site** or **Global** policy applies to this user).

Get all users in Lync site "Site01":

```
function Get-LyncSiteUser {
  [CmdletBinding()]
  param( [Parameter(Mandatory=$true)] [System.String] $LyncSiteName)
  $pools = @(Get-CsSite $LyncSiteName |
    Select-Object -ExpandProperty Pools)
  if ($pools.Count -gt 0) {
    Get-CsUser | Where-Object {$pools -contains $_.RegistrarPool} |
      Sort-Object RegistrarPool
  }
}
Get-LyncSiteUser Site01 |
  Format-Table DisplayName -GroupBy RegistrarPool
```

Managing Users

Enable AD user "User01" for Lync:

```
Enable-CsUser -Identity User01@example.com -RegistrarPool `
Pool01.example.com -SipAddress sip:User01@example.com
```

Disable an AD user for Lync functionality:

```
Disable-CsUser User01@example.com
```

(**Note:** this will keep the AD account, but it will be disabled for Lync. Lync attributes, including SIP address, will need to be reset if the account is re-enabled for Lync).

How are Lync users enabled & disabled for Federation, Public IM, and Internet (i.e. Remote) Access?

Managing these features is done by applying a Lync External User Access Policy. This policy can be set at a Global, Site, or User scope.

Modify the default Global External User Access Policy to enable Federation and Internet Access:

```
Set-CsExternalAccessPolicy -Identity Global -EnableFederationAccess `
$True -EnableOutsideAccess $True
```

Grant Public IM to all users in the Support department:

```
New-CsExternalAccessPolicy -Identity PublicIMEnabled `
-EnablePublicCloudAccess $True
Get-CsUser -LdapFilter "Department=Support" |
Grant-CsExternalAccessPolicy -Policy PublicIMEnabled
```

Enable all external access features for all users in Lync "Site01", including video with Public IM contacts:

```
New-CsExternalAccessPolicy -Identity site:Site01 -EnableFederationAccess `
$True -EnableOutsideAccess $True -EnablePublicCloudAccess $True `
-EnablePublicCloudAudioVideoAccess $True
```

(**Note:** because the new Site01 policy above is created at the Lync Site scope, it sets the external access properties for all Lync users in that site after creating the policy. No follow-on Grant-CsExternalAccessPolicy is required).

Remove an External Access Policy from "User01":

```
Get-CsUser sip:User01@example.com | Grant-CsExternalAccessPolicy
-PolicyName $Null
```

Enable a Lync "User01" for Enterprise Voice:

```
Set-CsUser User01@example.com -EnterpriseVoiceEnabled $True
```

Assign a Line URI to Lync user "User01":

```
Set-CsUser User01@example.com -LineUri tel:+12345
```

Assign the "Local" Dial Plan to "User01":

```
Grant-CsDialPlan User01@example.com -PolicyName "Local"
```

Assign the "Long Distance" Voice Policy to "User01":

```
Grant-CsVoicePolicy User01@example.com -PolicyName "Long Distance"
```

16 Common Lync User Properties

SipAddress	EnabledForFederation
Enabled	EnabledForInternetAccess
DisplayName	PublicNetworkEnabled
RegistrarPool	EnterpriseVoiceEnabled
LineURI	EnabledForRichPresence
Identity	ConferencingPolicy
WhenChanged	VoicePolicy
DialPlan	ClientPolicy

Working with Lync Clients

The previous Communicator group policy settings used in OCS are largely replaced with a new Lync **Client Policy**. A Group Policy is still needed for the client bootstrapping settings such the client sign-in settings. See http://technet.microsoft.com/en-us/library/gg398998.aspx for more information. The **Lync Client Policies cannot be configured in the Lync Control Panel**. It can be configured with the Lync Server PowerShell cmdlets.

Get all the settings in all the Client Policies:

```
Get-CsClientPolicy
```

Set the maximum number of contacts for Lync "User01" to 500:

```
New-CsClientPolicy -Identity MaxContactPolicy
-MaximumNumberofContacts 500
Grant-CsClientPolicy User01@example.com -PolicyName `
MaxContactPolicy
```

(**Note:** any existing client policy for User01 will be overwritten with the new policy. Also, see "Setting Maximum Number of Contacts" in the Gotcha's section for more information).

Show all Client Policies configured at the User scope:

```
Get-CsClientPolicy -Filter "tag:*"
```

Enable all users in Lync "Pool01" to transfer unencrypted files:

```
New-CsClientPolicy -Identity EnableUFileTransfer `
-EnableUnencryptedFileTransfer $True
Get-CsUser -Filter {RegistrarPool -eq "Pool01.example.com"} |
Grant-CsClientPolicy -PolicyName EnableUFileTransfer
```

Show which versions of clients (e.g. Communicator, Lync, etc…) are allowed to logon to Lync Server:

```
Get-CsClientVersionPolicy | Select-Object -ExpandProperty Rules
```

For more information about the Client Management Cmdlets:
http://technet.microsoft.com/en-us/library/gg398087.aspx

Working with Devices

Lync Server uses a Device Update Web service to distribute firmware updates to devices that run Lync 2010 Phone Edition.

Show the device update service configuration settings:

```
Get-CsDeviceUpdateConfiguration
```

Show the Lync 2010 Phone Edition configuration settings:

```
Get-CsUCPhoneConfiguration
```

Show all analog devices managed by Lync Server:

```
Get-CsAnalogDevice
```

Test if the user with phone number "+1 123 123 1234" can logon using a Lync 2010 Phone Edition-compatible device:

```
Test-CsPhoneBootstrap -PhoneOrExt "+11231231234" -Pin "1234"
```

For more information about Phone and Device Cmdlets:
http://technet.microsoft.com/en-us/library/gg415657.aspx

Working with the Address Book

The Lync Address book is a set of files holding AD user information that is used by Lync clients to search for contacts and normalize dialed numbers. User information from AD is synchronized into a Lync User Database by the Lync **User Replicator**. The Lync Address Book files are then generated from the data in the User Database, and synchronized every 5 minutes (by default).

Manually force the Address Book files on all Address Book servers to update from the Lync User Database:

```
Update-CsAddressBook
```

Get the Address Book configuration settings:

```
Get-CsAddressBookConfiguration
```

Change how often Address Book servers synchronizes with the data stored in the User database to 2 min:

```
Set-CsAddressBookConfiguration -SynchronizePollingInterval 00:02:00
```

Test if "User01" can download the Address Book from the Web Service in "Pool01":

```
$cred = Get-Credential "User01"
Test-CsAddressBookService -TargetFqdn Pool01.example.com `
-UserCredential $cred -UserSipAddress "sip:User01@example.com"
```

Show the User Replicator configuration settings in a Lync organization:

```
Get-CsUserReplicatorConfiguration
```

For more information on the Replication Cmdlets:
http://technet.microsoft.com/en-us/library/gg415677.aspx.

Working with the Topology (Sites, Pools, Servers, Computers)

Show information about all the computers used a Lync Server deployment:

```
Get-CsTopology
```

(**Note:** use the follow-on | **Select-Object -ExpandProperty <property name>** to expand and show the multi-valued properties such as Services).

Verify that a Lync deployment is functioning correctly at a global level:

```
Test-CsTopology
```

Show the Simple Url's in a Lync organization:

```
Get-CsSimpleUrlConfiguration | Select-Object -ExpandProperty SimpleUrl
```

Verify the replication status of the Central Management Store Configuration on all Lync Servers:

```
Get-CsManagementStoreReplicationStatus
```

Sites

Show all Central Lync sites and information about them:

```
Get-CsSite | Where-Object {$_.SiteType -eq "CentralSite"}
```

Pools

Show all Lync Pools and information about them:

```
Get-CsPool
```

Show all computers used in each Lync pool:

```
Get-CsPool | Select-Object -ExpandProperty Computers
```

Show all users homed on Lync "Pool01":

```
Get-CsUser | Where-Object {$_.RegistrarPool.FriendlyName -eq 'Pool01'}
```

Test that "User01" can logon to "Pool01":

```
$cred = Get-Credential "User01"
Test-CsRegistration -TargetFqdn Pool01.example.com `
-UserCredential $cred -UserSipAddress sip:User01.example.com
```

For more information on how to get all active connections on a registrar pool:
http://blogs.technet.com/b/csps/archive/2010/06/10/list-connections-to-registrar-pools.aspx

Server Roles

Show all Lync server roles, their configuration, and associated Lync service(s):

```
Get-CsService
```

Show all Lync servers of a particular role, and their configuration:

```
Get-CsService -<server role>
```

Lync server roles:

ApplicationServer	ManagementServer
ArchivingServer	MediationServer
CentralManagement	MonitoringServer
ConferencingServer	ProvisionServer
Director	PstnGateway
EdgeServer	UserServer

Show all Lync server roles and services in "Site01":

```
Get-CsService | Where-Object {$_.SiteID -eq "site:Site01"}
```

For more details on specific server roles cmdlets:

Address Book Server	http://technet.microsoft.com/en-ca/library/gg415643.aspx
Archiving and Monitoring	http://technet.microsoft.com/en-ca/library/gg415643.aspx
Edge Server	http://technet.microsoft.com/en-ca/library/gg415635.aspx
Database and Management Server	http://technet.microsoft.com/en-ca/library/gg415671.aspx
Registrar and Director	http://technet.microsoft.com/en-ca/library/gg415641.aspx
Web Server and Services	http://technet.microsoft.com/en-ca/library/gg415631.aspx

Computers

Show information about all the computers used in a Lync Server deployment:

```
Get-CsTopology | Select-Object -ExpandProperty Machines
```

Show all the network interfaces used on the computers in a Lync deployment:

```
Get-CsNetworkInterface
```

Get the configuration settings for computers running the Access Edge service (e.g. AllowFederatedUsers):

```
Get-CsAccessEdgeConfiguration
```

Report on the status of the Lync services running locally:

```
Test-CsComputer -Report c:\LocalLyncStatus.html
```

Working with Enterprise Voice

Lync Voice Policies enable and disable user call features such as Simultaneous Ring. They also determine Enterprise Voice call routing via the associated PSTN Usage(s).

Get the Enterprise voice policy settings for Lync user "User01" (including any setting for policies at the user, pool, or site level):

```
function Get-LyncUserVoiceConfiguration {
  [CmdletBinding()] param([Parameter(Position=0,Mandatory=$true)] [System.String]
  ${UserIdentity})
  if ($csUser = Get-CsUser $UserIdentity) {
    $site = (Get-CsSite | Where-Object `
      {$_.Pools -contains $csUser.RegistrarPool}).Identity
    $dialPlan = $csUser.DialPlan
    if (-not $dialPlan) {
      $dialPlansIds = @(Get-CsDialPlan | Select-Object -ExpandProperty Identity)
      if ($dialPlanIds -contains "Service:Registrar:$($csUser.RegistrarPool)") {
        $dialPlan = "Service:Registrar:$($csUser.RegistrarPool)"
      } elseif ($dialPlanIds -contains "site:$site") {
        $dialPlan = "Site:$site"
      } else {
        $dialPlan = 'Global'
      }
    }
    $voicePolicy = $csUser.VoicePolicy
    if (-not $voicePolicy) {
      $voicePolIds = @(Get-CsVoicePolicy | Select-Object -ExpandProperty Identity)
      if ($voicePolIds -contains "site:$site") {
        $voicePolicy = "Site:$site"
      } else {
        $voicePolicy = 'Global'
      }
    }
    $properties = @(
      @{'Name' = 'Enabled'
      'Expression' = {$_.EnterpriseVoiceEnabled}}
      @{'Name' = 'DialPlan'
      'Expression' = {$dialPlan}}
      @{'Name' = 'VoicePolicy'
      'Expression' = {$voicePolicy}}
      'LineURI'
    )
    $csUser | Select-Object $properties
  }
}
Get-LyncUserVoiceConfiguration User01
```

Show all the normalization rules configured across all Dial Plans in a Lync organization:

```
Get-CsDialPlan | Select-Object -ExpandProperty `
NormalizationRules
```

Test a dialed number against the Global Dial Plan. The matching normalization rule will be displayed along with the translated number:

```
Get-CsDialPlan -Identity "Global" | Test-CsDialPlan -DialedNumber `
14251231234 | Format-Table -Wrap
```

Show all the defined PSTN usages:

```
(Get-CsPstnUsage).Usage
```

Show all any Voice Routes that do not have a PSTN Gateway associated with it:

```
Get-CsVoiceRoute | Where-Object {$_.PstnGatewayList.Count -eq 0}
```

Remove the simultaneous ring feature in Lync user Voice Policy "BasicUserVoice":

```
Set-CsVoicePolicy BasicUserVoice -AllowSimulRing $false
```

Assign the "International" Voice Policy to all users in the Sales Organization Unit:

```
Get-CsUser -OU "ou=Sales,ou=Europe,dc=example,dc=com" |
Grant-CsVoicePolicy -PolicyName International
```

Test a PSTN outbound call to phone number "+1 425 123 1234" by "User01" using Lync "Pool01":

```
$cred = Get-Credential "User01"
Test-CsPstnOutboundCall -TargetFqdn Pool01.example.com `
-TargetPstnPhoneNumber "+14251231234" -UserSipAddress `
"sip:User01@example.com" -UserCredential $cred
```

Show the current inbound and outbound calls on Lync Mediation server "MedServer01":

```
Get-CsWindowsService -Name RTCMEDSRV -ComputerName `
MedServer01 | Format-List -Property ActivityLevel
```

Create a new collection of media settings that only use SRTP encryption between clients if it can be negotiated. You might want to do this as part of supporting audio and video with Public IM clients:

```
New-CsMediaConfiguration -Identity Site:Site01 -EncryptionLevel `
SupportEncryption
```

See http://technet.microsoft.com/en-us/library/gg415658.aspx for more details on the Enterprise Voice cmdlets.

Working with Certificates

The Lync Server Deployment Wizard is the only management UI that can be used to view and assign certificates. However, there are several Lync Management Shell cmdlets for managing certificates.

Show all certificates currently in use on the local Lync server and all of their properties, including the Subject Alternative Names:

```
Get-CsCertificate | Format-List -Property *
```

Show all certificates that expire before December 31, 2011:

```
Get-CsCertificate | Where-Object {$_.NotAfter -lt '12/1/2011'}
```

You can request and assign Lync certificates using the **Request-CsCertificate** or **Import-CsCertificate** and **Set-CsCertificate cmdlets**.

See http://technet.microsoft.com/en-us/library/gg398799.aspx for more information.

Gotchas

There are a few Lync PowerShell 'gotchas' to keep in mind:

Policy Scopes - Most Granular Wins

Most Lync policies and configurations can be applied at the following "Scopes":

1. **User** - applies to a user or group of users.
2. **Service** - applies to a service role (e.g. Registrar, User Services, or Pool.
3. **Site** - applies to a Lync site.
4. **Global** - applies through the entire Lync deployment. Policies and Configurations have a default Global instance.

Where multiple scopes apply, **the most specific scope takes precedence**. The order of precedence is User, Service, Site, and Global.

For more details see http://blogs.technet.com/b/csps/archive/2010/06/06/refscopes.aspx.

Setting the Maximum Number of Contacts

The maximum number of contacts allowed by a user can be set at 4 different scopes: **Global, Site, Service**, and the **User** level. The cmdlets for each scope are:

- The **Global** setting is read/set through the **Get/Set-CsUserServicesConfiguration** cmdlet.
- The **Site** and **Service** settings are set through using the **New-CsUserServicesConfiguration** cmdlet with an -Identity of **site**:<site_name> for Site, and **service**:<pool_name> for the service level scope.
- The **User** scope is controlled through the New-CsClientPolicy and Grant-CsClientPolicy cmdlets.

The **User** policy setting **takes precedence** as long as the maximum number configured in it is less than the maximum set in the Global, Site, or Service levels.

Useful Links

Microsoft TechNet:

- Lync 2010 Management Shell
- Lync 2010 Cmdlets by Category
- Lync 2010 PowerShell Blog

Free PowerShell community, forums, administrative and scripting/debugging tools:
http://PowerGUI.org
Lync Tips, Tricks, & Insight: http://www.InsideLync.com

Author

Curtis Johnstone http://www.InsideLync.com & http://www.InsideOCS.com

Contributors

Kirk Munro http://poshoholic.com