Basic Powershell Commands to MECM

**Basic Powershell Commands :**

**1.Below Command Displays Site Server Name, SiteCode And Site Version**

Get-CMSite | Select ServerName, SiteCode, Version

**2.Below Command Displays Roles installed on Primary Site Server (MECMPRIMARY.LAB.LOCAL)**

Get-CMSiteRole -SiteSystemServerName mecmprimary.lab.local | Select RoleName

**3. Below Command Displays the list of users present in MECM Database**

Get-CMUser | Select Name

**4. Below Command Displays the list of Devices present in MECM Database**

Get-CMDevice | Select Name

**5. Below Command Displays the list of Packages in MECM**

Get-CMPackage -fast | Select Name

**6. Below Command Displays the list of Applications in MECM**

Get-CMApplication | Select LocalizedDisplayName

**7. Below Command Displays the list of Boundaries and their value in MECM**

Get-CMBoundary | Select DisplayName, Value

**8. Below Command Displays the list of Boundary Groups in MECM**

Get-CMBoundaryGroup | Select Name

**9. Below Command Displays the list of Compliance Baseline Name and its compliance result in MECM**

Get-CMBaseline | Select LocalizedDisplayName, ComplianceCount, NonComplianceCount

**10. Below Command Displays the list of All deployments made in MECM**

Get-CMDeployment | Select ApplicationName, CollectionName, DeploymentTime

**Powershell Commands To Create User Collections**

New-CMUserCollection -Name 'Canada Users' -LimitingCollectionName 'All Users'

New-CMUserCollection -Name 'Brazil Users' -LimitingCollectionName 'All Users'

**Powershell Commands To Create Device Collections**

New-cmdevicecollection -Name 'Brazil Computers' -LimitingCollectionName 'All Systems'

New-cmdevicecollection -Name 'Canada' -LimitingCollectionName 'All Systems'

**Create Device collections in sccm using powershell and move it to a specific folder**

**Step 1: Create Folders**

New-Item -Name 'Google Chrome' -Path 'PS1:\DeviceCollection'

New-Item -Name 'Firefox' -Path 'PS1:\DeviceCollection'

New-Item -Name '7zip' -Path 'PS1:\DeviceCollection'

New-Item -Name 'VLC Media Player' -Path 'PS1:\DeviceCollection'

New-Item -Name 'Visio' -Path 'PS1:\DeviceCollection'

**Step 2: Create Device Collections**

New-CMDeviceCollection -Name 'Google\_Chrome\_Device\_Collection' -LimitingCollectionName 'All Systems'

New-CMDeviceCollection -Name 'Firefox\_Device\_Collection' -LimitingCollectionName 'All Systems'

New-CMDeviceCollection -Name '7zip\_Device\_Collection' -LimitingCollectionName 'All Systems'

New-CMDeviceCollection -Name 'VLC\_Media\_Player\_Device\_Collection' -LimitingCollectionName 'All Systems'

New-CMDeviceCollection -Name 'Visio\_Device\_Collection' -LimitingCollectionName 'All Systems'

**Step 3: Define Variables**

$Collection1 = Get-CMDeviceCollection -Name 'Google\_Chrome\_Device\_Collection'

$Collection2 = Get-CMDeviceCollection -Name 'Firefox\_Device\_Collection'

$Collection3 = Get-CMDeviceCollection -Name '7zip\_Device\_Collection'

$Collection5 = Get-CMDeviceCollection -Name 'VLC\_Media\_Player\_Device\_Collection'

$Collection6 = Get-CMDeviceCollection -Name 'Visio\_Device\_Collection'

**Step 4: Move collections into specified folders**

Move-CMObject -InputObject $Collection1 -FolderPath 'PS1:\DeviceCollection\Google Chrome'

Move-CMObject -InputObject $Collection2 -FolderPath 'PS1:\DeviceCollection\Firefox'

Move-CMObject -InputObject $Collection3 -FolderPath 'PS1:\DeviceCollection\7zip'

Move-CMObject -InputObject $Collection5 -FolderPath 'PS1:\DeviceCollection\VLC Media Player'

Move-CMObject -InputObject $Collection6 -FolderPath 'PS1:\DeviceCollection\Visio'

**Create Direct Rule Device collections in sccm using PowerShell**

$collectiondir = "C:\Users\mecmadm\Desktop\report"

$collectionname = "Canada Laptops"

$computers = Get-Content "C:\Users\mecmadm\Desktop\report\PC\_List.txt"

New-CMDeviceCollection -Name $collectionname -LimitingCollectionName "All Systems"

foreach($computer in $computers) {

try {

Add-CMDeviceCollectionDirectMembershipRule -CollectionName $collectionname -ResourceId $(get-cmdevice -Name $computer).ResourceID

}

catch {

"Invalid client or direct membership rule may already exist: $computer" | Out-File "$collectiondir\$collectionname`\_invalid.log" -Append

}

}

**Create Query Rule Device collections in sccm using PowerShell**

**1. Define Variables**

$collectiondir = "C:\Users\mecmadm\Desktop\report"

$collectionname = "All Windows 10 Computes"

$Query = "select \* from SMS\_R\_System where SMS\_R\_System.OperatingSystemNameandVersion like '%workstation%'"

**2. Create New Collection**

New-CMDeviceCollection -Name $collectionname -LimitingCollectionName "All Systems"

**3. Add members in the collection using query rule method**

Add-CMDeviceCollectionQueryMembershipRule -CollectionName $collectionname -QueryExpression $Query -RuleName 'Query1'

**Packaging Cisco Anyconnect with Java as a pre-requisite using Powershell App Deployment Toolkit (PSADT)**

**Install Java :**

$setup = Start-Process "\mecmlab\source\Applications\JRE\jre-8u192-windows-x64.exe" -ArgumentList "/s" -Wait -NoNewWindow

**Uninstall Java :**

$app = Get-WmiObject -Class win32\_Product | Where-Object {$\_.Name -eq "Java 8 Update 221 (64-bit)"}

$app.uninstall()

**Create And Deploy Application In Using Powershell**

**1. Define Variables**

$appname = "Application Name"

$publisher = "Vendor"

$version = "Version"

$location = "Application Location"

$collname = "Collection Name"

$description = "Description"

$icon = "Icon Location"

$dp = "DP Name"

$install = "Install Command Line"

$uninstall = "Uninstall Command Line"

**2. Create New Application**

New-CMApplication -Name $appname -Description $description -SoftwareVersion $version -Publisher $publisher -AutoInstall $true -IconLocationFile $icon -Verbose

**3. Add Deployment Type**

Add-CMMsiDeploymentType -ApplicationName $appname -ContentLocation $location -InstallationBehaviorType InstallForSystem -InstallCommand $install -UninstallCommand $uninstall

**4. Distribute The App To DP**

Start-CMContentDistribution -ApplicationName $appname -DistributionPointName $dp -Verbose

**5. Deployment**

New-CMApplicationDeployment -CollectionName $collname -Name $appname -DeployAction Install -DeployPurpose Available -UserNotification DisplayAll -AvailableDateTime (Get-Date) -TimeBaseOn LocalTime -Verbose