



Getting Started

This chapter contains the following sections:

- [Connecting to Cisco IMC, on page 1](#)
- [Default Cisco IMC, on page 2](#)
- [Connect to Multiple Cisco IMCs, on page 2](#)
- [Credentials To and From a File, on page 3](#)
- [SSL Handling, on page 4](#)
- [Aliases, on page 4](#)

Connecting to Cisco IMC

Step 1 From the desktop shortcut, launch IMC PowerTool.

Step 2 View all cmdlets, functions, and aliases supported by Cisco IMC PowerTool, using the following cmdlets:

```
Get-Command -Module Cisco.Imc
Get-Command -Module Cisco.Imc | group CommandType
Get-Command -Module Cisco.Imc | measure
```

Step 3 Connect to a Cisco IMC, using the following cmdlets:

```
$handle = Connect-Imc <ip or hostname> -NotDefault
```

After logging on, by default, the Cisco IMC handle is added to the default Cisco IMC list, unless the `-NotDefault` option is specified. Every cmdlet that operates on a Cisco IMC takes the `-Imc` parameter, where the handle can be specified.

Step 4 Connect to a Cisco IMC using a proxy, using the following cmdlets:

```
$proxy = New-Object System.Net.WebProxy
$proxy.Address = "http:\\<url>:<port>"
$proxy.UseDefaultCredentials = $false
$proxy.Credentials = New-Object System.Net.NetworkCredential("<user name>", "<password>")
$handle = Connect-Imc <ip or hostname> -Proxy $proxy
```

Step 5 Use the following cmdlets:

- a) Get the consolidated status information from the Cisco IMC.

```
Get-ImcStatus -Imc $handle
```

- b) Get the inventory summary of the Cisco IMC.

```
Get-ImcRackUnit -Imc $handle
```

- c) Disconnect.

```
Disconnect-Imc -Imc $handle
```

Default Cisco IMC

If a no handle or name is specified, the Cisco IMC handle is added to a DefaultImc server list unless the `-Imc` parameter is specified. The first cmdlet in the pipeline operates on the default Cisco IMC list.

Connect to Cisco IMC

```
Connect-Imc <ip or hostname>
```

Get the default Cisco IMC

```
Get-UcsPsSession
```

Get the status information and Cisco IMC version

```
Get-ImcStatus
```

Get Cisco IMC server details

```
Get-ImcRackUnit
```

Enable HTTP on Cisco IMC

```
Get-ImcHttp | Set-ImcHttp -AdminState enabled
```

Disable HTTP on Cisco IMC

```
Get-ImcHttp | Set-ImcHttp -AdminState disabled
```

Disconnect Cisco IMC

```
Disconnect-Imc
```

Connect to Multiple Cisco IMCs

When you specify multiple handles, Cisco IMC PowerTool cmdlets can work with multiple Cisco IMCs.

Use the following cmdlets to connect to multiple IMCs:

Connecting to a Cisco IMC:

```
$handle1 = Connect-Imc <ip1> -NotDefault
$handle2 = Connect-Imc <ip2> -NotDefault
Get-ImcStatus -Imc $handle1,$handle2
Disconnect-Imc -Imc $handle1,$handle2
```

By default, multiple Cisco IMC handles are not allowed in DefaultImc. You can override this restriction by using the **Set-UcsPowerToolConfiguration** cmdlet.

```
Get-UcsPowerToolConfiguration
Set-UcsPowerToolConfiguration -SupportMultipleDefaultUcs $true
Connect-Imc <ip1>
Connect-Imc <ip2>
Get-ImcStatus
Disconnect-Imc
```

Connecting to Multiple Cisco IMC:

You can use the credentials which you used for connecting to a Cisco IMC.

```
$user = "<username>"
$password = "<password>" |
ConvertTo-SecureString -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential($user, $password)
$servers = @("<Imc1>", "<Imc2>", "<Imc3>")
Connect-Imc $servers -Credential $cred
```

Credentials To and From a File

```
Connect-Imc <ip1>
Connect-Imc <ip2>
```

Credentials can be stored in a file. The stored credentials are encrypted with a specified key.

```
Export-UcsPssession -LiteralPath C:\work\labs.xml
Disconnect-Imc
```

A login can be initiated from credentials stored in a file.

```
Connect-Imc -LiteralPath C:\work\labs.xml
```

Specify proxy while logging in with credentials stored in a file.

```
$proxy = New-Object System.Net.WebProxy
$proxy.Address = "http://<url>:<port>"
$proxy.UseDefaultCredentials = $false
$proxy.Credentials = New-Object System.Net.NetworkCredential("<user name>", "<password>")
Connect-Imc -LiteralPath C:\work\lab.xml -Proxy $proxy
```

Log in to an extra system and add the credentials to the file.

```
Connect-Imc <ip3>
Export-UcsPssession -Path C:\work\lab.xml -Merge
```

SSL Handling

When you connect to a Cisco IMC, the server does not recognize the valid certificates. The connection depends on `InvalidCertificateAction`. `InvalidCertificateAction` is set to `Ignore` by default. By default, Cisco IMC PowerTool is configured to establish the connection without a valid certificate.

You can override this setting by using the `Set-UcsPowerToolConfiguration` cmdlet.

```
Get-UcsPowerToolConfiguration
Set-UcsPowerToolConfiguration -InvalidCertificateAction Fail
```

The following table describes the options for checking the validity of the certificate:

Options	Description
Fail	The cmdlet does not establish connection if the certificate is not valid.
Ignore	The cmdlet establishes a connection without considering that the certificate is invalid.
Default	(Windows default) The cmdlet establishes a connection if the certificate is valid.

Aliases

Some aliases are predefined for convenience. To view the list of all aliases, run the following cmdlet:

```
gal | ? {$_.Name -like "*-Imc*" } | select Name
```

The following table lists the aliases and the corresponding cmdlets:

Alias	Cmdlet
Clear-ImcK mipLogin	Get-ImcK mipServerLogin Set-ImcK mipServerLogin -AdminAction clear
Clear-ImcK mipServer	Get-ImcK mipServer Set-ImcK mipServer -AdminAction clear
Remove-ImcRootCACertificate	Get-ImcK mipManagement Set- ImcK mipManagement -AdminAction delete-root-ca-certificate
Remove-ImcClientCertificate	Get-ImcK mipManagement Set- ImcK mipManagement -AdminAction delete-client-certificate

Alias	Cmdlet
Remove-ImcClientPrivateKey	Get-ImcKmpManagement Set- ImcKmpManagement -AdminAction delete-client-private-key
Enable-ImcBiosProfile	Get-ImcBiosProfile Set-ImcBiosProfile -AdminAction activate
Remove-ImcBiosProfile	Get-ImcBiosProfile Set-ImcBiosProfile -AdminAction delete
Backup-ImcBiosProfile	Get-ImcBiosProfileManagement Set-ImcBiosProfileManagement -AdminAction backup
Clear-ImcOneTimePrecisionBoot Device	Get-ImcOneTimePrecisionBoot Device Set ImcOneTimePrecisionBoot Device -AdminAction clear-one-time-boot-device
Reset-ImcStorageController	Get-ImcStorageController Set-ImcStorageController -AdminAction delete-all-vds-reset-pds
Clear-ImcBootDrive	Get-ImcStorageController Set-ImcStorageController -AdminAction clear-boot-drive
Clear-ImcForeignConfig	Get-ImcStorageController Set-ImcStorageController -AdminAction clear-foreign-config
Disable-ImcJbod	Get-ImcStorageController Set-ImcStorageController -AdminAction disable-jbod
Enable-ImcJbod	Get-ImcStorageController Set-ImcStorageController -AdminAction enable-jbod
Get-ImcTtyLog	Get-ImcStorageController Set-ImcStorageController -AdminAction get-tty-log
Import-ImcForeignConfig	Get-ImcStorageController Set-ImcStorageController -AdminAction import-foreign-config
Add-ImcMo	Add-ImcManagedObject
Disable-ImcLocatorLed	Set-ImcLocatorLed -AdminState off
Enable-ImcLocatorLed	Set-ImcLocatorLed -AdminState on
Enable-ImcPidCatalog	Set-ImcActivatePIDCatalog -AdminState trigger
Get-ImcMo	Get-ImcManagedObject
Remove-ImcLdapCertificate	Set-ImcLdapCACertificate -AdminAction delete-ca-certificate

Alias	Cmdlet
Remove-ImcMo	Remove-ImcManagedObject
Reset-ImcServer	Set-ImcRackUnit -AdminPower hard-reset-immediate
Reset-ImcEventFileters	Set-ImcEventManager -AdminAction reset-event-filters
Restart-ImcServer	Set-ImcRackUnit -AdminPower cycle-immediate
Set-ImcMo	Set-ImcManagedObject
Start-ImcServer	Set-ImcRackUnit -AdminPower up
Stop-ImcServer	Set-ImcRackUnit -AdminPower soft-shut-down
Invoke-ImcPowerCharacterization	Set-ImcPowerBudget -AdminAction start-power-char
Reset-ImcPowerProfile	Set-ImcPowerBudget -AdminAction reset-power-profile-default
Test-ImcLdapBinding	Set-ImcLdapCACertificate -AdminAction test-ldap-binding