Cisco UCS Director PowerShell Agent Installation and Configuration Guide, Release 6.5

First Published: 2017-07-11

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Preface

Audience

This guide is intended primarily for data center administrators who use Cisco UCS Director and who have responsibilities and expertise in one or more of the following:

- Server administration
- Storage administration
- Network administration
- Network security
- Virtualization and virtual machines

Conventions

Text Type	Indication
GUI elements	GUI elements such as tab titles, area names, and field labels appear in **this font**. Main titles such as window, dialog box, and wizard titles appear in **this font**.
Document titles	Document titles appear in *this font*.
TUI elements	In a Text-based User Interface, text the system displays appears in `this font`.

Text Type	Indication
System output	Terminal sessions and information that the system displays appear in `this font`.
CLI commands	CLI command keywords appear in **this font**. Variables in a CLI command appear in *this font*.
[ ]	Elements in square brackets are optional.
{x | y | z}	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the document.

Caution Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

Tip Means *the following information will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Warning

IMPORTANT SAFETY INSTRUCTIONS

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Related Documentation

Cisco UCS Director Documentation Roadmap

For a complete list of Cisco UCS Director documentation, see the *Cisco UCS Director Documentation Roadmap* available at the following URL: http://www.cisco.com/en/US/docs/unified_computing/ucs/ucs-director/doc-roadmap/b_UCSDirectorDocRoadmap.html.

Cisco UCS Documentation Roadmaps

For a complete list of all B-Series documentation, see the *Cisco UCS B-Series Servers Documentation Roadmap* available at the following URL: http://www.cisco.com/go/unifiedcomputing/b-series-doc.

For a complete list of all C-Series documentation, see the *Cisco UCS C-Series Servers Documentation Roadmap* available at the following URL: http://www.cisco.com/go/unifiedcomputing/c-series-doc.

Note

The *Cisco UCS B-Series Servers Documentation Roadmap* includes links to documentation for Cisco UCS Manager and Cisco UCS Central. The *Cisco UCS C-Series Servers Documentation Roadmap* includes links to documentation for Cisco Integrated Management Controller.

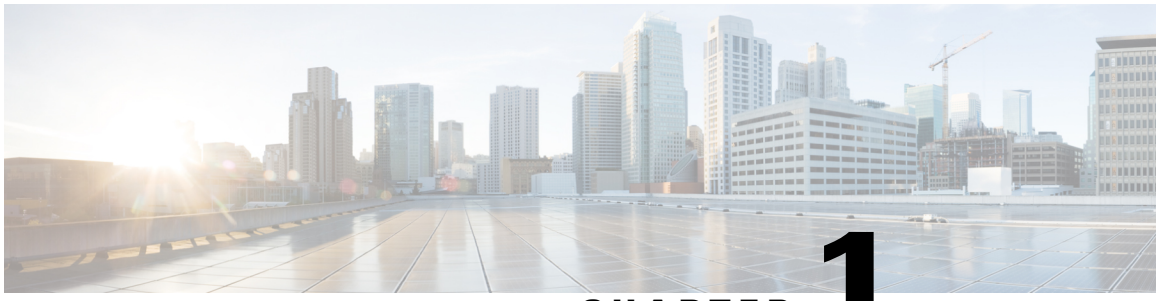
Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to ucs-director-docfeedback@cisco.com. We appreciate your feedback.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the What's New in Cisco Product Documentation RSS feed. RSS feeds are a free service.



CHAPTER 1

New and Changed Information for this Release

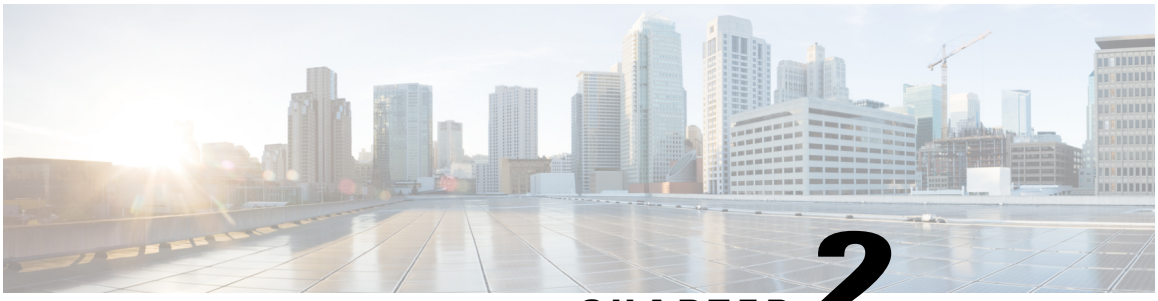
This chapter contains the following section:

New and Changed Information for This Release

The following table provides an overview of the significant changes to this guide for this current release. The table does not provide an exhaustive list of all changes made to this guide or of all new features in this release.

Table 1: New and Changed Information for Release 6.5

Feature	Description	Where Documented
Authentication Mechanisms	This feature allows you to choose an authentication mechanism for your environment. Earlier, you only had a default option.	Authentication Mechanisms, on page 11
Execute Native PowerShell Command task	The new Execute Native PowerShell Command task helps you to execute PowerShell scripts on the PowerShell Agent server. This task overcomes the limitation where some of the third- party cmdlets are not executable in remote sessions.	Task Library See also .Limitations of Execute Native PowerShell Command Task, on page 17



CHAPTER 2

Overview

This chapter contains the following sections:

Cisco UCS Director PowerShell Agent

Cisco UCS Director PowerShell Agent is a lightweight Microsoft Windows service application that acts as an interface layer between Cisco UCS Director and the Windows machine.

You can download a PowerShell Agent and install it on a Windows machine that has WinRM enabled. After you have started PowerShell Agent on a Windows machine, establish a connection between the PowerShell Agent and Cisco UCS Director. This connectivity enables you to execute PowerShell scripts to automate infrastructure configuration through Cisco UCS Director.

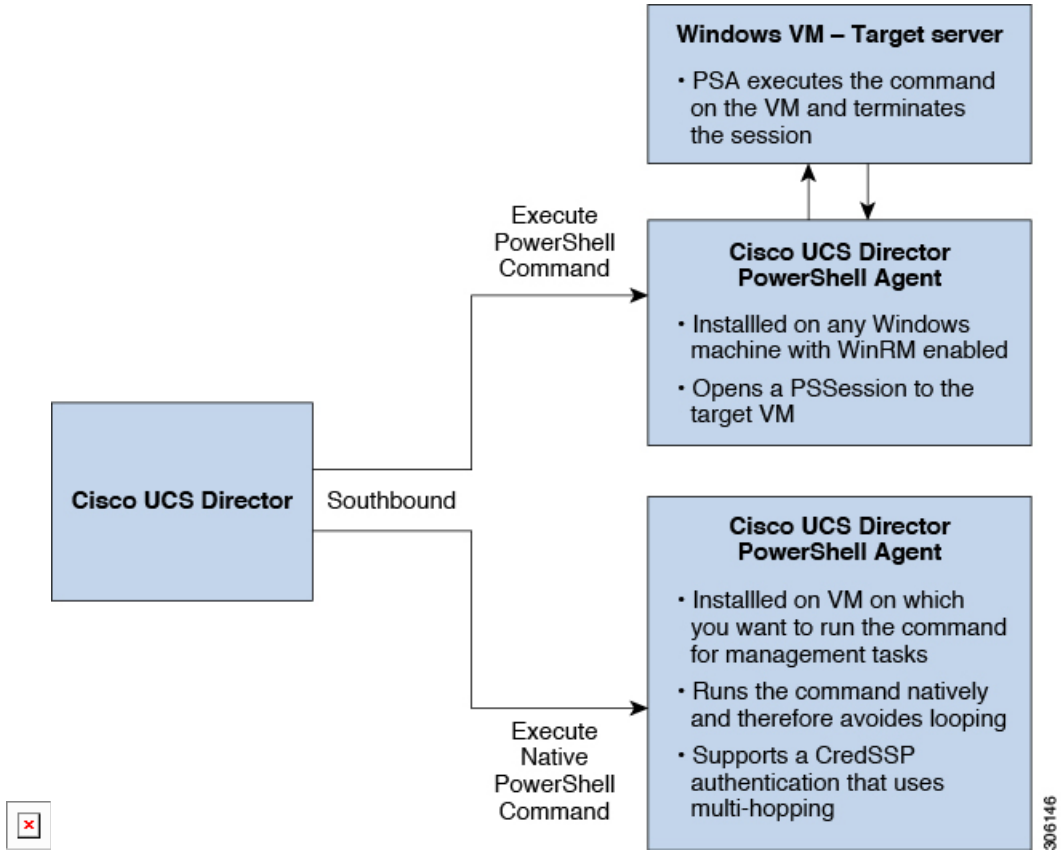
PowerShell Agent initiates a remote PowerShell session (PSSession) on the target server to run PowerShell commands. The target server is any Windows machine that is included in the WinRM configuration and that PowerShell Agent can access through the default WinRM port.

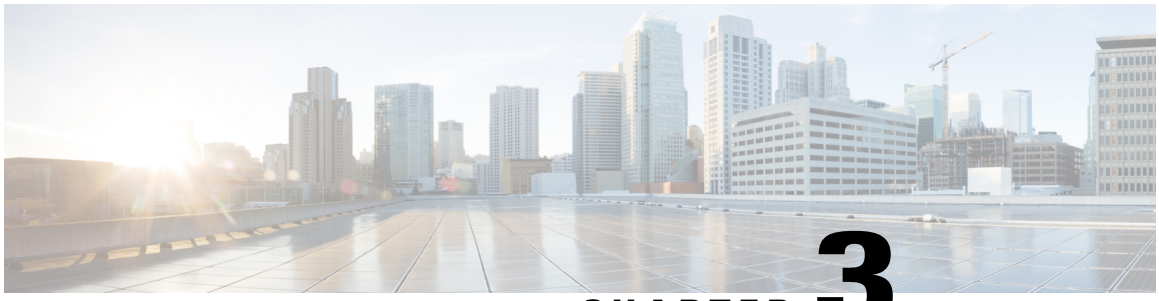
When a PowerShell command is executed through a Cisco UCS Director workflow task, the following occurs:

1. Cisco UCS Director sends the HTTP encoded command to PowerShell Agent.
2. PowerShell Agent establishes WSMAN connections to the remote machines and then executes the commands on them.
3. The output of the command is converted to XML and sent back to PowerShell Agent.
4. PowerShell Agent terminates the connection to the target server.
5. PowerShell Agent returns the output to Cisco UCS Director as the payload in HTTP response.

6 Other Cisco UCS Director workflow tasks can parse the returned PowerShell object information and use it as one or more variables.

Figure 1: Overview of Cisco UCS Director PowerShell Agent





CHAPTER 3

Installing Cisco UCS Director PowerShell Agent

This chapter contains the following sections:

Prerequisites

Before you install Cisco UCS Director PowerShell Agent, make sure that the following software has been installed on the machine:

- Supported Windows operating system.
- Supported .NET Framework (Full Package).
- Configure WinRM on any device that the PowerShell Agent communicates with.

For more information about the system requirements and supported software, see the Compatibility Matrix for Cisco UCS Director.

Enabling WinRM and WinRS

The PowerShell Agent executes cmdlets and scripts on the target server in a PowerShell remote session. To accept remote PowerShell commands, enable Windows Remote Management (WinRM), change the startup type to *Automatic*, create a "listener" to respond to WinRS commands, and start the service on each computer you want to work with. This provides connectivity to Windows Remote Shell (WinRS), the client side of WS-Management protocol.

To enable WinRM, run a configuration command. The command creates a "listener" and also opens an exception for WinRM in Windows Firewall.

Step 1 Open a command prompt, and enter the command:
winrm quickconfig

You receive the following output:

```
WinRM is not set up to allow remote access to this machine for management.
The following changes must be made:

Set the WinRM service type to delayed auto start.
Start the WinRM service.
Create a WinRM listener on HTTP://* to accept WS-Man requests to any IP on this machine.
Enable the WinRM firewall exception.
Make these changes [y/n]?
```

Step 2 Enter **y**.
You receive the following output:

```
Make these changes [y/n]? y

WinRM has been updated for remote management.

WinRM service type changed successfully.
WinRM service started.
Created a WinRM listener on HTTP://* to accept WS-Man requests to any IP on this machine.
WinRM firewall exception enabled.
```

Step 3 Verify that WinRM is enabled by executing the following command:
get-service winrm

Step 4 Configure the value `" * "` in the TrustedHosts table of WinRM by executing the following command:
winrm set winrm/config/client @{TrustedHosts="*"}

When you are working with computers in workgroups or home groups, either use HTTPS as the transport or add the remote machine to the TrustedHosts configuration settings to enable authentication.

Note Configuring WinRM over HTTP or HTTPS depends on the requirements of the specific environment. HTTPS is only necessary if a secure connection is required. For more details, see https://blogs.technet.microsoft.com/meamcs/2012/02/24/how-to-force-winrm-to-listen-interfaces-over-https/.

What to Do Next

Make sure that the domain account used for connecting to a target server belongs to the local administrator group.

Configuring the Firewall

The PowerShell Agent listens for incoming requests on port 43891. This is the default port. Configure your machine's firewall to allow incoming TCP requests on any other port of your choice.

Cisco UCS Director services create an access key, that is used for secure communication on that port. Copy that key from the main Cisco UCS Director interface and enter it into the PowerShell Agent host.

Downloading Cisco UCS Director PowerShell Agent

Download the installer for PowerShell Agent from Cisco UCS Director to your native Windows machine.

Step 1 Choose **Administration** > **Virtual Accounts**.

Step 2 Click **PowerShell Agents**.

Step 3 Click **Download Installer**.

Step 4 Review the list of installation requirements on the **Download Agent Installer** page. Ensure that you have them available on the Windows machine where you plan to install the PowerShell Agent.

Step 5 Click **Submit**.
The `PSASetup.exe` file is downloaded to your native Windows machine default download folder.

What to Do Next

Install Cisco UCS Director PowerShell Agent on your Windows machine.

Installing Cisco UCS Director PowerShell Agent

Install the PowerShell Agent on your native Windows machine to enable the Cisco PSA service.

Installing a newer version of the PowerShell Agent requires that you uninstall the older version first. To remove the older version of PowerShell Agent, stop the Cisco PSA Service first and then uninstall the agent.

Note If you do not install the current version of PowerShell Agent for Cisco UCS Director on the Windows machine, some tasks or options on the **PowerShell Agents** tab are not available.

Before You Begin

- You need system administrator privileges to complete this task.
- Enable WinRM.

- Configure Firewall.

Step 1 If necessary, copy the `PSASetup.exe` file that you downloaded from Cisco UCS Director to your target Windows machine.

Step 2 Double-click the **PSASetup.exe** file.

Step 3 In the **Cisco PSA Service - InstallShield Wizard** screen, click **Next**.

Step 4 In the **Ready to install the Program** screen, click **Install**.
The **Installing Cisco PSA Service** screen displays during the installation. When the installation is complete, the **InstallShield Wizard Completed** message is displayed.

Step 5 Click **Finish**.
The PowerShell Agent is installed to the `C:\Program Files (x86)\Cisco Systems\Cisco PSA Service` folder. This folder is referred to as `%AGENT_INSTALL_FOLDER%` in the remainder of the document.

Step 6 Verify that the Cisco PSA Service is running on the Windows machine by checking the Resource Monitor.

Updating Ports and Authentication Properties for PowerShell Agent

By default, the PowerShell Agent uses port 43891 and it also uses a predefined authentication key to communicate with Cisco UCS Director. You can change these values by modifying the following file:

`%AGENT_INSTALL_FOLDER%/props/properties.xml`.

Before You Begin

Download and install the Cisco UCS Director PowerShell Agent service on your target server.

Step 1 Use a text editor to open the *properties.xml* file.

Step 2 Edit the authKey entry.

Step 3 Edit the serverPort entry.

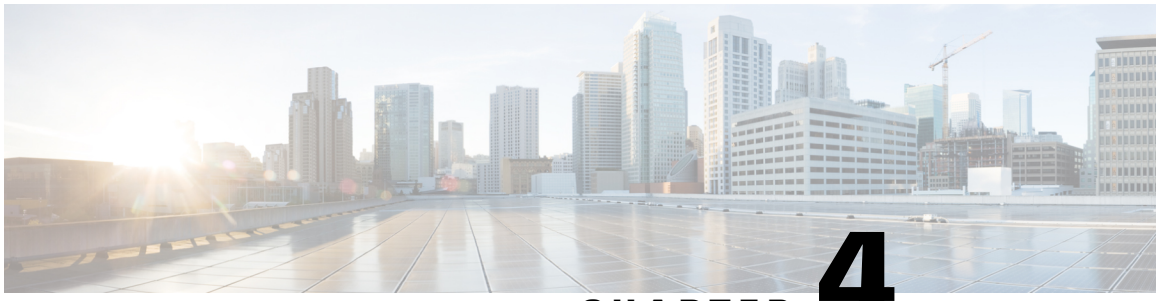
Step 4 Save the file.

Note The port number and access key values must match the values that are specified in the PowerShell Agent. Cisco UCS Director cannot communicate with the PowerShell Agent if the entries do not match.

Step 5 Restart the Cisco UCS Director PowerShell Agent service if you modify any of the properties in the *properties.xml* file. This can be done from the Services snap-in (**Start > Services.msc**)

What to Do Next

If you change the default port, configure the firewall to match the new port.



CHAPTER 4

Configuring Cisco UCS Director PowerShell Agent

This chapter contains the following sections:

Adding PowerShell Agent to Cisco UCS Director

After PowerShell Agent is installed and running, add it to Cisco UCS Director. Ensure to set up the virtual account (for example, an SCVMM account) to use the PowerShell Agent for inventory collection and other management functions.

Step 1 Choose **Administration** > **Virtual Accounts**.

Step 2 On the **Virtual Accounts** page, click **PowerShell Agents**.

Step 3 Click **Add**.

Step 4 On the **Add Agent** screen, complete the following fields.

Name	Description
Agent name field	The name of the PowerShell Agent.
Agent Address field	The IP address or FQDN (Fully Qualified domain name) of the PowerShell Agent.

Name	Description
Agent Access Port field	The port assigned to the PowerShell Agent. **Note** The default port address is 43891. These values are pre-populated with default values. The PowerShell Agent is pre-configured to use these values and UCS Director must have matching values. These values can be changed on the Agent, but UCS Director must always use them as well.
Access key field	The default access key assigned to the PowerShell Agent. **Note** These values are prepopulated with default values. The PowerShell agent is preconfigured to use these values and Cisco UCS Director must have matching values. These values can be changed on the PowerShell Agent, but Cisco UCS Director must always use them as well.
Description field	The description of the PowerShell Agent.

Step 5 Click **Submit**.

What to Do Next

Verify connectivity between Cisco UCS Director and the PowerShell Agent.

Verifying Connectivity with Cisco UCS Director

After the PowerShell Agent is added, you can check the connectivity between Cisco UCS Director and the PowerShell Agent.

Step 1 Choose **Administration** > **Virtual Accounts**.

Step 2 On the **Virtual Accounts** page, click **PowerShell Agents**.

Step 3 From the **More Actions** drop-down list, choose **Test Connection**.
Cisco UCS Director displays a success message if it can communicate with the PowerShell Agent.

If Cisco UCS Director cannot communicate with the PowerShell Agent, see Troubleshooting Connectivity with Cisco UCS Director, on page 11.

What to Do Next

Execute the Cisco UCS Director PowerShell command.

Troubleshooting Connectivity with Cisco UCS Director

Problem

You can experience a failed test connection with Cisco UCS Director. This problem can occur even though you successfully installed and configured the PowerShell Agent, and there is no issue with the network connectivity between PowerShell Agent and Cisco UCS Director.

Note This problem can happen with Windows Server 2012 R2 or other versions that use advanced cipher suites for https communication.

Symptom

Check the PowerShell Agent log files in the PowerShell Agent server, for an **SSPI failed with inner exception** error. See sample error message:

2014-08-20 14:44:16,832 [6] ERROR cuic.ClientConnection[null] - Exception: A call to SSPI failed, see inner exception.
2014-08-2014:44:16,832 [6] DEBUG cuic.ClientConnection[null] - Inner exception: The message received was unexpected or badly formatted.
2014-08-2014:44:16,832 [6] DEBUG cuic.ClientConnection[null] - Authentication failed - closing the connection.

Cause

The test connection fails because of the Microsoft update, in which, new TLS cipher suites are added and cipher suite priorities are changed in Windows RT 8.1, Windows 8.1, and Windows Server 2012 R2. See Microsoft kb article 2929281 for further information on this update.

Solution

To resolve the problem, modify the SSL cipher suite group policy setting. Follow the listed steps:

1. At a command prompt, enter `gpedit.msc` to open your group policy editor.
2. Expand **Computer Configuration** > **Administrative Templates** > **Network**, and then click **SSL Configuration Settings**.
3. Under **SSL Configuration Settings**, click the **SSL Cipher Suite Order** setting.
4. In the **SSL Cipher Suite Order** pane, scroll to the bottom of the pane.
5. Follow the instructions labeled **How to modify this setting**.

It is necessary to restart the computer after modifying this setting for the changes to take effect.

Authentication Mechanisms

After you have added the PowerShell Agent to Cisco UCS Director, you can set the authentication mechanism by creating a workflow with Execute PowerShell Command task.

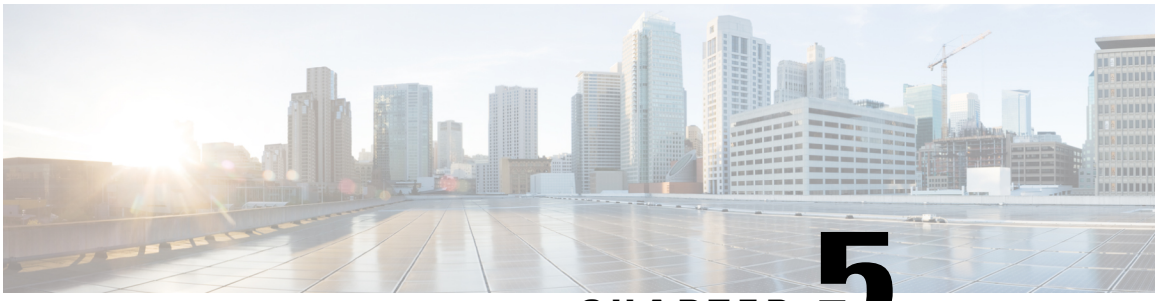
The **Execute PowerShell Command Task** establishes a remote PowerShell session from PowerShell Agent to the target server to execute commands on that server. A **Default** authentication mechanism is currently used to set up the session. With this release, support is also extended to the following types of authentication mechanisms:

- **Basic Authentication—**Simple mechanism to transmit username and password to a web server or target machine in clear text.
- **Kerberos Authentication—**Mutual authentication process that uses encrypted keys between a client and a server machine. This protocol is selected to authenticate a domain account such that both the user identity and server identity are guaranteed without sending of any reusable credentials.
- **Negotiate Authentication—**Both the client and the server compute a session key from the user password without ever exchanging the password itself. It is selected for local computer accounts and is best suited for intranet web authentication.
- **Negotiate Authentication with Implicit Credentials—**Assigns an SSL certificate to a target server to guarantee both the user and the server identity. A client trusted Certificate Authority issues the SSL certificate.
- **CredSSP Authentication—**Intended for environments where Kerberos delegation cannot be used. To use CredSSP authentication, delegate the PowerShell Agent as a CredSSP client for the target machine.

Note Multi-hop support in Windows Remote Management (WinRM) uses CredSSP for authentication. Since PowerShell is built on top of WinRM, you can use CredSSP to perform multi-hop authentication.

When you add a new workflow for the Execute PowerShell Command Task, one of the input fields is **Authentication Mechanism**. It provides you an option to choose the type of authentication you wish to set-up for the remote session.

For detailed steps on how to execute the task and set your authentication between the PowerShell Agent and target server, see Executing PowerShell Commands, on page 15.



CHAPTER 5

Executing PowerShell Agent Commands

This chapter contains the following sections:

Cisco UCS Director Orchestrator Workflow and PowerShell Command

Cisco UCS Director Orchestrator automates complex tasks by organizing them into workflows. Once built and validated, these workflows perform the same way every time, no matter who runs the workflows. These tasks encompass a wide variety of supported Cisco and non-Cisco hardware and software data center components.

A workflow, for example, can consist of the following Orchestrator elements:

- Start icon
- Task icons (like the Execute PowerShell Command)
- Both Completed (Success) task and Competed (Failed) task icons

For more information about workflows, see Cisco UCS Director Orchestration Guide.

PowerShell commands are used for executing workflows on a target server. Cisco UCS Director offers the following two types of command tasks:

- **Execute PowerShell Command Task**
- **Execute Native PowerShell Command Task**

Execute PowerShell Command Task

The **Execute PowerShell Command Task** can run PowerShell scripts only on a remote server. When a command is executed, PowerShell Agent opens a remote PowerShell connection (PSSession) to the target server. Scripts are then executed on the target server. The connection is closed once the execution is complete.

Limitations of the Execute PowerShell Command Task

1. In certain scenarios, PowerShell cmdlets are not executable and the script fails. This failure occurs because opening the remote session (PSSession) does not provide a complete PowerShell interactive desktop capability. As a result, the scripts are run in an environment that is different from the native PowerShell Agent console. For example, some Windows Active Directory cmdlets cannot be executed under PSSession unless they are run from a device that has a specific Windows Active Directory role associated with it.
2. PowerShell scripts may also fail if your environment has a multi-hop delegation and you have not enabled the CredSSP protocol in your infrastructure. This failure occurs because the scripts that are executed remotely cannot connect to other Windows machines.

Example of Execute PowerShell Command Task Inputs:

- PowerShell Agent to be used for executing the script.
- Target Server's credentials (IP address, username and password, and domain)
- Commands or Scripts of up to 64 kb.

When you execute the workflow in Cisco UCS Director, you are prompted to enter the PowerShell Agent commands to run on a target server. Use a ";" to separate multiple commands (for example, Hostname; Get-Process). Cisco UCS Director runs the commands against the target server and displays the output as an XML string in a service request log window.

See Executing PowerShell Commands for detailed steps.

Execute Native PowerShell Command Task

The **Execute Native PowerShell Command Task** creates a native PowerShell instance on a Windows machine and executes the scripts natively on the PowerShell Agent. As a result, the target scripts are not sent over a remote WinRM session for execution. They are run locally on the PowerShell Agent. This feature allows you to bypass the limitations encountered when the scripts are run in a PSSesion using the **Execute PowerShell Command Task**.

The Execute Native PowerShell Command task simulates the PowerShell Console on a Windows machine. As a result, the command runs natively in a PowerShell CLI session. All the functionality offered by the PowerShell console is available through this task.

Limitations of the Execute Native PowerShell Command Task

1. Certain windows commands are not supported when the output format is set to JSON and as a result a task may fail. There are no issues if the output format is set to XML.

   Run the following command to convert the output format to XML:

   ```
   dir| ConvertTo-Xml -As String
   ```

2 The **Write-Error** cmdlet causes the task to fail because this cmdlet is not compatible with the one on the console.

3 Certain cmdlets, such as **Enter-PSSession** and **Write-Error**, require an interactive shell fail with the error message `method not implemented`. You can use **Invoke-Command** if the cmdlet Enter-PSSession fails.

See Executing PowerShell Commands for detailed steps.

Executing PowerShell Commands

Open a web browser and log on to Cisco UCS Director to execute the PowerShell commands.

Step 1 Choose **Orchestration**.

Step 2 On the **Orchestration** page, click **Workflows**.

Step 3 Click **Add**.

Step 4 Complete the fields for the **Add Workflow** wizard.

Step 5 Click **Submit**.

Step 6 On the **Available Tasks** screen, select the **Execute PowerShell Command** task. Drag and drop the task in the Workflow Designer pane.

Step 7 Double click the **Execute PowerShell Command**.

Step 8 On the **Task Information** screen, leave the default values. Click **Next**.

Step 9 On the **User Input Mapping** screen, check the **Map to User Input** box if you want prompts for any of the task values during workflow execution. Click **Next**.

Step 10 On the Task Inputs screen, complete the following fields.

Name	Description
Label field	Enter a name for the task.
PowerShell Agent drop-down list	Select the PowerShell agent to be used for executing the script.
Target Machine IP field	Enter the target machine IP address. Provide the DNS or NetBIOS name for Kerberos authentication.
User ID field	Specify the local admin user for basic authentication.
Password field	Enter the password.
Domain field	(Optional) Enter the domain name of the target server.
Authentication Mechanism drop-down list	Select the authentication type. See Authentication Mechanisms, on page 11
Commands/Script	Provide a script for any task that you want to run later (like add a domain, GET-Process, and so on).

Name	Description
Commands/Rollback Script	The Commands and Scripts to be executed on the PowerShell Agent if the executed scripts fail. This is an optional field.
Output Format drop-down list	Choose the output format of the PowerShell script. Choices are XML or JSON.
Depth drop-down list	Enter the level to which the contained objects are shown in the XML/JSON output.
Maximum Wait Time drop-down list	Enter the time (in minutes) for which the task waits for the scripts to be executed.

Note In the **Commands/Script** field, certain character sequences, like `/` and `$` may not work because the PowerShell Agent runs an **Invoke-Command** cmdlet with the remote PSSession and there are some limitations in sending special characters to that cmdlet.

Step 11 Click **Submit**.

Step 12 Click **Validate** to verify the new workflow.

Step 13 Click **Execute**.
The **Command Output** window displays the execution results.

Example: Setting Up PowerShell Agent and Running a Test Task

The following example outlines how you can set up PowerShell Agent on a Windows server and run a test task.

Step 1 Create a Microsoft Windows Server 2008 R2 or 2012 R2 VM.

Step 2 Make sure that the VM has the required .NET Framework and Windows PowerShell versions.

Step 3 Open a web browser and log on to Cisco UCS Director.

Step 4 Choose **Administration** > **Virtual Accounts**.

Step 5 On the **Virtual Accounts** page, click **PowerShell Agents**.

Step 6 Click **Download Installer** and install the PowerShell Agent. See Downloading Cisco UCS Director PowerShell Agent, on page 7

Step 7 In Windows Firewall, open the port that has been configured for the PowerShell Agent (the default port is 43891).

Step 8 Open PowerShell and run the following commands:
Enable-PSRemoting -Force

Set-Item WSMan:\localhost\Client\TrustedHosts -Value "" -Force*

Restart-Service WinRM

Set-ExecutionPolicy unrestricted -Force

Step 9 Log on to Cisco UCS Director again and run the Execute the PowerShell command. See Executing PowerShell Commands, on page 15.

Limitations of Execute Native PowerShell Command Task

The Execute Native PowerShell Command task helps you to execute PowerShell scripts on the PowerShell Agent server. This task overcomes the limitations of the some of the third party cmdlets, which might not execute in remote sessions.

Note To run PowerShell scripts from a remote server, use the Execute PowerShell Command task.

Though the **Execute Native PowerShell Command** task simulates the **PowerShell Console** on a Windows server to the closest extent possible, there are a few limitations:

- The Write-Error cmdlet does not work similarly to the one on the PowerShell console. The Write-Error cmdlet causes the task to fail.
- Certain cmdlets (for example, Enter-PSSession and Write-Host) that require an interactive shell do not work. Such cmdlets fail with the following error message:

  ```
  Method not implemented
  ```

  As the Enter-PSSession cmdlet does not work for establishing remote sessions, the workaround is to use Invoke-Command.