



Executing PowerShell Agent Commands

This chapter contains the following sections:

- [Cisco UCS Director Orchestrator Workflow and PowerShell Command, on page 1](#)
- [Execute PowerShell Command Task, on page 2](#)
- [Execute Native PowerShell Command Task, on page 2](#)
- [Executing PowerShell Commands, on page 3](#)
- [Example: Setting Up PowerShell Agent and Running a Test Task, on page 5](#)
- [Limitations of Execute Native PowerShell Command Task, on page 5](#)
- [Execute Parse PowerShell Output Command Task, on page 6](#)

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 Completed (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 PSSession 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.
4. When executing the **ExecutingNativePowerShellComand** task in certain scenarios such as **Import-Module** cmdlets on SCVMM 2012, you might get the following error message:

```
Mixed mode assembly is built against version 'v2.0.50727' of the runtime and cannot be loaded in the 4.0 runtime without additional configuration information.
```

In such scenarios, it is recommended to use the **ExecutePowershellCommand** task instead of the **ExecutingNativePowerShellComand** task.

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.

Name	Description
Domain field	(Optional) Enter the domain name of the target server.
Authentication Mechanism drop-down list	Select the authentication type. See Authentication Mechanisms
Commands/Script	Provide a script for any task that you want to run later (like add a domain, GET-Process, and so on).
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 <ul style="list-style-type: none"> • After waiting for the specified duration, the task ends, independent of the state of the script execution. • Cisco recommends that you configure a time-out at the script level, shorter than Maximum Wait Time, to avoid creating stale objects. For example, you can use the PowerShell job feature to define the time-out period.

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.

To hide script content in the PSA log file, use the `<ucsd:mask> </ucsd:mask>` tag. For example to hide a password 'Pass123' in the PSA logs, enclose the password in the tag as `<ucsd:mask>Pass123</ucsd:mask>`.

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 above.
- 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 PowerShell Agent from Cisco UCS Director](#)
- 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 3](#).
- 

## 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.

## Execute Parse PowerShell Output Command Task

The **Parse PowerShell Output Command** task requires input in XML or JSON values only. You can map this task to the **Execute PowerShell Command Task** or the **Execute Native PowerShell Command Task**. The output of these two tasks is in an XML format. When this XML output is passed as input for the **Parse PowerShell Output Command** task, the resulting output is an XML response which provides values in an individual parameter format.

### Limitations of the Execute Parse PowerShell Output Command Task

- The output generated from this task is dynamic. So, to view the output fields, you must first run this task with sample input values.
- This task processes the input only if the XML or JSON values are provided in a specific format.

Following is a sample output of the **Execute PowerShell Command Task** for **\$PsVersionTable.PSVersion** in XML and JSON formats.

XML Format:

```
[Output: POWERSHELL_COMMAND_RESULT = <?xml version='1.0'?><Objects><Object
Type='System.Version'>4.0</Object></Objects>]
```

JSON Format:

```
[Output: POWERSHELL_NATIVE_COMMAND_RESULT = {
 'PSVersion': {
 'Major': 5,
 'Minor': 1,
 'Build': 17763,
 'Revision': 316,
 'MajorRevision': 0,
 'MinorRevision': 316
 },
 'PSEdition': 'Desktop',
 'PSCompatibleVersions': [
 '1.0',
 '2.0',
 '3.0',
 '4.0',
 '5.0',
 '5.1.17763.316'
],
 'BuildVersion': {
 'Major': 10,
 'Minor': 0,
 'Build': 17763,
 'Revision': 316,
 'MajorRevision': 0,
 'MinorRevision': 316
 }
}]
```