Cisco UCS Director VM Guest Customization 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

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 Features and Changed Behavior in Cisco UCS Director, Release 6.5.

Feature	Description	Where Documented
Execute VM Command task enhancements	The task supports rollback and the option to capture command output and exit codes.	Execute VM Command Task, on page 35
Execute VIX Scripts task support for Guest Operations APIs	The task supports the option to use Guest Operations APIs instead of VIX APIs.	Support for Guest Operations APIs, on page 7 Examples with Guest Operations API Option Enabled, on page 18
New Guest Operations task	The Guest Operations task supports the existing VIX task functionality with the ability to upload and execute files on the guest.	Guest Operations Task, on page 41

CHAPTER 2

Overview

This chapter contains the following sections:

VM Guest Customization Options

To customize guest operating systems on a VM, Cisco UCS Director provides you the following options:

- The Execute VIX Scripts task, the Execute VM Commands task, and the Guest Operations tasks from the Cisco UCS Director library tasks, which help execute VIX scripts or commands on the VM.
- The " userAPIExecuteVIXScript" API and the "EXECUTE_VIX_SCRIPT" API, which help execute VIX scripts on the VM.

Prerequisites

Before you consider customizing your guest operating systems through Cisco UCS Director, read through the following prerequisites:

- Install the latest VMware tools on the vCenter.
- Power on the VM that has the guest operating system installed.
- Make sure the latest VMware tools are installed and running on the guest VM.
- Obtain the administrator credentials for the guest VM.
- Make sure the ESX/ESXi host is reachable from UCS Director.

Escape Characters

An escape sequence is a backslash (\) followed by a character. Escape characters are used to specify actions such as inserting a new line and a tab. The following table lists the Java escape sequences:

Escape Sequence	Inserts
\t	Tab
\b	Backspace
\n	New line
\r	Carriage return
\f	Formfeed
\'	Single quotation mark
\"	Double quotation mark
\\	Backslash
\$	Dollar variable
\\|	Pipe to send output of one command to other in text
\+	Plus character
\^	Caret character
\\&	Ampersand character

When an escape sequence is encountered in a command, the compiler interprets it accordingly. For example, to use quotes within quotes, use the escape sequence (\") on the interior quotes. To print `SQL instance name is "ucsd_instance1"`, enter the command as follows:

```
echo "SQL instance name is \"ucsd_instance1\""
```

If the command contains a backslash (\), add one more '\' character as prefix to the '\' character to send the correct command to the VM. If the command includes a UCSD variable (${variable}), use four backslashes '\\\\' instead of two.

Examples:

```
C:\\WINDOWS\\system32\\cmd.exe mkdir C:\\test1     // two backslashes

C:\\\\sqlinstall\\\\install.cmd ${SQL_INSTANCE_NAME}        // four backslashes,
                                                            // as UCS Director variables
are used in the command
```

Status Codes

The following table lists some of the VIX status codes:

Code	Description
0	Command successfully executed
1	Unknown error
127	Exception thrown – Command not found
9009	Exception thrown - Not recognized as an operable program or command
3014	Insufficient permissions in host operating system
3015	Authentication failure or insufficient permissions in guest operating system

For a complete list of VIX status codes, refer to VIX Error Codes documentation on the VMware Support site.

After execution of commands, you can check the Cisco UCS Director workflow logs to see whether the VIX script has successfully executed the command. On successful execution, the workflow log shows the status code as 0. For a failed task or passing of irrelevant commands, the VIX script shows a status code other than 0.

Note In the Cisco UCS Director workflow logs, you see that the ERROR_STATUS_MESSAGE is empty when EXIT_STATUS_CODE = 0 and EXIT_STATUS_CODE = 1. Other Exit Status Codes, however, display proper error messages.

After the task is completed on the VM, you can verify the status code in the VM as well. For a Linux VM, verify the status code by entering the command **echo $?**. For a Windows VM, verify the status code by entering **echo %ERRORLEVEL%**.

CHAPTER 3

Executing VIX Scripts on Guest VMs

This chapter contains the following sections:

Executing VIX Scripts in Cisco UCS Director

A VIX script executes Linux and Windows commands on a Guest VM. Cisco UCS Director provides the following methods to execute VIX scripts:

- The Execute VIX Scripts task, which executes the Linux or Windows command on a Guest OS.
- The userAPIExecuteVIXScript API (JSON based) and the EXECUTE_VIX_SCRIPT API (XML based)

Note However, the Rest APIs have the following limitations:

  - The response displays only the status code. The response does not display the output of the valid command.
  - To view error messages, you must use the userAPIExecuteVIXScriptWithOutput command.

Support for Guest Operations APIs

The VIX API is deprecated in VMware vSphere versions after VMware vSphere version 6.0. Starting with VMware vSphere version 5.0, VIX APIs are integrated into the VMware vSphere Web Services SDK.

The Execute VIX Script task supports the option to use the Guest Operations API (VMware vSphere SDK). To execute Linux and Windows command using the VMware vSphere SDK, check **Invoke Guest Operation API**. When the Guest Operations API option is enabled, the Execute VIX Script automatically uses the latest Guest OS API layer instead of calling the VIX APIs.

The VMware vCenter version 6.5 VIX command is supported using the new Guest Operations API. If you are using vCenter version 6.5, you must select this option and update any existing workflows.

The VMware vCenter version 6.0 VIX command supports the use of both VIX API and Guest Operations APIs.

Note The **Invoke Guest Operation API** check box addresses the upgrade path from VIX APIs to Guest Operations APIs without modifying the commands or scripts used within the Execute VIX Script task.

By default, the Guest Operations API is not enabled. A warning message appears in the SR logs stating that the VIX API will be deprecated in future releases of VMware vSphere.

You can view log details for a service request.

See the Cisco UCS Director Administration Guide.

Examples with Output Display Option Enabled

The Output Display option is available in the Execute VIX Script task. You can choose to enable the Output Display option whenever you use the Execute VIX Scripts task.

When the Output Display option is enabled for running Windows commands on a VM, a batch file and an output file are created in the **/opt/infra/vix_temp_files** directory. The batch file contains the command to be executed on the selected VM. After the command is executed on the VM, an output file is created. The output file contains the status code of the batch file execution.

When the Output Display option is enabled for running Linux commands on a VM, just an output file is created. The output file captures the status code of the batch file execution.

The following examples illustrate how you can use the Execute VIX Script Task for running commands on Windows and Linux VMs, with the Output Display option enabled.

Executing a Single-Line Windows Command

In this example, we execute a single-line Windows command using the Execute VIX Script task, with the Output Display option enabled. When the Output Display option is enabled for running Windows commands on a VM, a batch file is created. The batch file contains the command to be executed on the selected VM. An output file captures the status code of the batch execution.

Enter the following information in the Execute VIX Scripts task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. In the **Script** field, enter the following sample Windows command:

   ```
   C:\\WINDOWS\\system32\\cmd.exe mkdir C:\\test_Dir_9867
   ```
4. Check **Output Display**.

After executing the task, you see the following information in the log:

```
Service Request ID: 536

Mar 21, 2016 10:03:00 UTC Request Submitted

Mar 21, 2016 10:03:03 UTC Executing workflow item number 1

Mar 21, 2016 10:03:03 UTC Completed workflow item number 0, with status completed

Mar 21, 2016 10:03:09 UTC Executing workflow item number 2

Mar 21, 2016 10:03:00 UTC Trigger context executeWorkflowStep called

Mar 21, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Mar 21, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Mar 21, 2016 10:03:09 UTC VIX API will be deprecated in future release.

Mar 21, 2016 10:03:13 UTC  [VIXActionHandler] - account details VMName : Win_2k12 , Host:
172.31.234.46, OS Type: microsoft windows server 2012 (64-bit), ExitCode :0

Mar 21, 2016 10:03:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 3 seconds

Mar 21, 2016 10:03:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Mar 21, 2016 10:03:13 UTC [Local Input: Select VM = 3312)

Mar 21, 2016 10:03:13 UTC [Local Input: Credential type = Login]

Mar 21, 2016 10:03:13 UTC [Local Input: Login =  administrator]

Mar 21, 2016 10:03:13 UTC [Local Input: Password = "masked-value")

Mar 21, 2016 10:03:13 UTC [Local Input: Script - C:\\WINDOWS\\system32\\cmd.exe mkdir
C:\\test_Dir_9867]

Mar 21, 2016 10:03:13 UTC [Local Input: Undo Script =  ]

Mar 21, 2016 10:03:13 UTC (Local Input: Error Codes = ]

Mar 21, 2016 10:03:13 UTC [Local Input: Output display = true]

Mar 21, 2016 10:03:13 UTC (Output: EXIT_STATUS_CODE = 0]

Mar 21, 2016 10:03:13 UTC (Output: ERROR_STATUS_MESSAGE = ]

Mar 21, 2016 10:03:13 UTC Completed workflow item number 1, with status completed
```

As seen in the log, the script returns status code 0, which means that the task has been executed successfully.

A batch file is created in the following location:

```
/opt/infra/vix_temp_files/vixbatch-03_21_2016_10_03_09_835.bat
```

The content of the batch file is as follows:

```
@echo off
mkdir C:\\test_Dir_9867
echo status_code: %ERRORLEVEL%
```

The output file is created in the following location:

```
 /opt/infra/vix_temp_files/vixoutput-03_21_2016_10_03_09_835.txt
```

The content of the output file is as follows:

```
status_code: 0
```

Executing a Multiline Windows Command

In this example, we execute a multiline Windows command using the Execute VIX Script task, with the Output Display option enabled. When the Output Display option is enabled for running Windows commands on a VM, a batch file is created. The batch file contains the command to be executed on the selected VM. An output file captures the status code of the batch file execution.

Enter the following information in the Execute VIX Scripts task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. In the **Script** field, enter the following sample Windows command:

```
C:\\Windows\\system32\\cmd.exe echo 1sttt file > c:\\One1.txt && echo 2nddd file >
c:\\Three.txt
```

Note Use && for multiline commands. Multiline commands do not support &.

4. Check **Output Display**.

After executing the task, you see the following information in the SR log:

```
Apr 06, 2016 12:06:00 UTC Request Submitted

Apr 06, 2016 12:06:03 UTC Executing workflow item number 1

Apr 06, 2016 12:06:03 UTC Completed workflow item number 0, with status completed

Apr 06, 2016 12:06:09 UTC Executing workflow item number 2

Apr 06, 2016 12:06:00 UTC Trigger context executeWorkflowStep called

Apr 06, 2016 12:06:09 UTC Executing custom action VIX2008 (Execute VIX Script)

Apr 06, 2016 12:06:09 UTC Executing custom action VIX2008 (Execute VIX Script)

Apr 06, 2016 12:06:09 UTC VIX API will be deprecated in future release.

Apr 06, 2016 12:06:13 UTC  [VIXActionHandler] - account details VMName : vm-DEF-SR119 ,
Host: 172.31.234.85, OS Type: microsoft windows server 2012 (64-bit), ExitCode :0

Apr 06, 2016 12:06:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 3 seconds

Apr 06, 2016 12:06:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 06, 2016 12:06:13 UTC [Local Input: Select VM = 392)

Apr 06, 2016 12:06:13 UTC [Local Input: Credential type = Login]

Apr 06, 2016 12:06:13 UTC [Local Input: Login =  administrator]

Apr 06, 2016 12:06:13 UTC [Local Input: Password = "masked-value")

Apr 06, 2016 12:06:13 UTC [Local Input: Script - C:\\WINDOWS\\system32\\cmd.exe /c fsutil
file x.txt | echo.hi > C:\\util.txt]

Apr 06, 2016 12:06:13 UTC [Local Input: Undo Script =  ]

Apr 06, 2016 12:06:13 UTC (Local Input: Error Codes = ]
```

```
Apr 06, 2016 12:06:13 UTC [Local Input: Output display = true]

Apr 06, 2016 12:06:13 UTC (Output: EXIT_STATUS_CODE = 0]

Apr 06, 2016 12:06:13 UTC (Output: ERROR_STATUS_MESSAGE = ]

Apr 06, 2016 12:06:13 UTC Completed workflow item number 1, with status completed

Apr 06, 2016 12:06:13 UTC Completed workflow item number 3

Apr 06, 2016 12:06:13 UTC Completed workflow item number 2, with status completed
```

As seen in the log, the script returns status code 0, which means that the task has been executed successfully.

A batch file is created in the following location:

```
/opt/infra/vix_temp_files/vixbatch-04_07_2016_02_01_34_231.bat
```

The content of the batch file is as follows:

```
@echo off
echo 1sttt file > c:\\One1.txt && echo 2nddd file > c:\\Three.txt
echo status_code: %ERRORLEVEL%
```

The output file is created in the following location:

```
 /opt/infra/vix_temp_files/vixoutput-04_07_2016_02_01_34_231.txt
```

The content of the output file is as follows:

```
status_code: 0
```

Executing a Windows Command with Error Codes Specified

In this example, we execute a single-line Windows command using the Execute VIX Script task, with the Output Display option enabled and an error code specified. When the Output Display option is enabled for running Windows commands on a VM, a batch file is created. The batch file contains the command to be executed on the selected VM. An output file captures the status code of the batch file execution.

Enter the following information in the Execute VIX Scripts task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. In the **Script** field, enter the following invalid Windows command:

   ```
   C:\\WINDOWS\\system32\\cmd.exe mkkdir C:\\test_error_code
   ```
4. In the **Error Codes** field, enter 9009.
5. Check **Output Display**.

As the user has passed an invalid command and specified the error code (9009), the task checks against the user-provided error code and fails the task.

```
Service Request ID: 486

Apr 21, 2016 09:05:00 UTC Request Submitted

Apr 21, 2016 09:05:03 UTC Executing workflow item number 1

Apr 21, 2016 09:05:03 UTC Completed workflow item number 0, with status completed

Apr 21, 2016 09:05:09 UTC Executing workflow item number 2
```

```
Apr 21, 2016 09:05:00 UTC Trigger context executeWorkflowStep called

Apr 21, 2016 09:05:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 21, 2016 09:05:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 21, 2016 09:05:09 UTC VIX API will be deprecated in future release.

Apr 21, 2016 09:05:13 UTC  [VIXActionHandler] - account details VMName : Win_2k12 , Host:
172.31.234.46, OS Type: microsoft windows server 2012 (64-bit), ExitCode :0

Apr 21, 2016 09:05:13 UTC Action Execute VIX Script: Task failed with Error Code: 9009 and
 status message: "mkkdir"
not recognized as an internal or external command, operable program or batch file

Apr 21, 2016 09:05:13 UTC Task: (vix_script (Execute VIX Script) failed with error - Task
failed with Error Code: 9009 and status message: 'mkkdir'
not recognized as an internal or external command, operable program or batch file. ,
selectedContext=<None>

Apr 21, 2016 09:05:13 UTC Task#1 (vix_script (Execute VIX Script)) failed after 6 seconds

Apr 21, 2016 09:05:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 21, 2016 09:05:13 UTC [Local Input: Select VM = 3312)

Apr 21, 2016 09:05:13 UTC [Local Input: Credential type = Login]

Apr 21, 2016 09:05:13 UTC [Local Input: Login =  administrator]

Apr 21, 2016 09:05:13 UTC [Local Input: Password = "masked-value")

Apr 21, 2016 09:05:13 UTC [Local Input: Script - C:\\WINDOWS\\system32\\cmd.exe mkdir
C:\\test_error_code]

Apr 21, 2016 09:05:13 UTC [Local Input: Undo Script =  ]

Apr 21, 2016 09:05:13 UTC (Local Input: Error Codes = 9009]

Apr 21, 2016 09:05:13 UTC [Local Input: Output display = true]

Apr 21, 2016 09:05:13 UTC (Output: EXIT_STATUS_CODE = 9009]

Apr 21, 2016 09:05:13 UTC (Output: ERROR_STATUS_MESSAGE = 'mkkdir' not recognized as an
internal or external command, operable program or batch file.]

Apr 21, 2016 09:05:13 UTC Completed workflow item number 1, with status failed
```

As seen in the log, the script returns status code 9009, which was the value provided in the Error Code field.

If the command is invalid, but the error code is not provided, the task does not check the guest error code against the user field error code. So the task is shown as successful. The task returns an appropriate guest error code as follows:

```
Service Request ID: 523

Apr 21, 2016 10:05:00 UTC Request Submitted

Apr 21, 2016 10:05:03 UTC Executing workflow item number 1

Apr 21, 2016 10:05:03 UTC Completed workflow item number 0, with status completed

Apr 21, 2016 10:05:09 UTC Executing workflow item number 2

Apr 21, 2016 10:05:00 UTC Trigger context executeWorkflowStep called

Apr 21, 2016 10:05:09 UTC Executing custom action vix_script (Execute VIX Script)
```

```
Apr 21, 2016 10:05:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 21, 2016 09:05:09 UTC VIX API will be deprecated in future release.

Apr 21, 2016 10:05:13 UTC  [VIXActionHandler] - account details VMName : Win_2k12 , Host:
172.31.234.46, OS Type: microsoft windows server 2012 (64-bit), ExitCode :0

Apr 21, 2016 10:05:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
after 3 seconds

Apr 21, 2016 10:05:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 21, 2016 10:05:13 UTC [Local Input: Select VM = 3312)

Apr 21, 2016 10:05:13 UTC [Local Input: Credential type = Login]

Apr 21, 2016 10:05:13 UTC [Local Input: Login =  administrator]

Apr 21, 2016 10:05:13 UTC [Local Input: Password = "masked-value")

Apr 21, 2016 10:05:13 UTC [Local Input: Script - C:\\WINDOWS\\system32\\cmd.exe mkkdir
C:\\test_error_code]

Apr 21, 2016 10:05:13 UTC [Local Input: Undo Script =  ]

Apr 21, 2016 10:05:13 UTC (Local Input: Error Codes = ]

Apr 21, 2016 10:05:13 UTC [Local Input: Output display = true]

Apr 21, 2016 10:05:13 UTC (Output: EXIT_STATUS_CODE = 9009]

Apr 21, 2016 10:05:13 UTC (Output: ERROR_STATUS_MESSAGE = 'mkkdir' is not recognized as an
 internal or external command, operable program or batch file.]

Apr 21, 2016 10:05:13 UTC Completed workflow item number 1, with status completed
```

Executing a Single-Line Linux Command

In this example, we execute a single-line Linux command using the Execute VIX Script task, with the Output Display option enabled. When the Output Display option is enabled for running Linux commands on a VM, an output file is created. The output file captures the status code of the batch execution.

Enter the following information in the Execute VIX Scripts task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. In the **Script** field, enter the following sample Linux command:

   ```
   /bin/mkdir /tmp/testdir
   ```

4. Check **Output Display**.

After executing the task, you see the following information in the SR log:

```
Apr 22, 2016 12:06:00 UTC Request Submitted

Apr 22, 2016 12:06:03 UTC Executing workflow item number 1

Apr 22, 2016 12:06:03 UTC Completed workflow item number 0, with status completed

Apr 22, 2016 12:06:09 UTC Executing workflow item number 2
```

```
Apr 22, 2016 12:06:00 UTC Trigger context executeWorkflowStep called

Apr 22, 2016 12:06:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 22, 2016 12:06:09 UTC Executing custom action Vix_script (Execute VIX Script)

Apr 22, 2016 12:06:09 UTC VIX API will be deprecated in future release.

Apr 22, 2016 12:06:13 UTC  [VIXActionHandler] - account details VMName : vm-DEF-SR119 ,
Host: 172.31.234.85, OS Type: red hat enterprise linux 4(32-bit), ExitCode :0

Apr 22, 2016 12:06:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 4 seconds

Apr 22, 2016 12:06:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 22, 2016 12:06:13 UTC [Local Input: Select VM = 9801)

Apr 22, 2016 12:06:13 UTC [Local Input: Credential type = Login]

Apr 22, 2016 12:06:13 UTC [Local Input: Login =  administrator]

Apr 22, 2016 12:06:13 UTC [Local Input: Password = "masked-value")

Apr 22, 2016 12:06:13 UTC [Local Input: Script - /bin/mkdir /tmp/testdir]

Apr 22, 2016 12:06:13 UTC [Local Input: Undo Script =  ]

Apr 22, 2016 12:06:13 UTC (Local Input: Error Codes = ]

Apr 22, 2016 12:06:13 UTC [Local Input: Output display = true]

Apr 22, 2016 12:06:13 UTC (Output: EXIT_STATUS_CODE = 0]

Apr 22, 2016 12:06:13 UTC (Output: ERROR_STATUS_MESSAGE = ]

Apr 22, 2016 12:06:13 UTC Completed workflow item number 1, with status completed
```

As seen in the log, the script returns status code 0, which means that the task has been executed successfully.

Unlike Windows, a separate shell script is not created. Only an output file is created. The output file is created in the following location:

```
/opt/infra/vix_temp_files/vixoutput-04_07_2016_10_03_09_835.txt
```

The content of the output file is as follows:

```
status_code: 0
```

Examples with Output Display Option Disabled

When the Output Display option in the Execute VIX Scripts task is unchecked, the commands are directly executed on the Windows or Linux VM. A batch file or an output file is not created.

The following examples illustrate how you can use the Execute VIX Script Task for running commands on Windows and Linux VMs, with Output Display option not enabled.

Executing A Single-Line Windows Command

In this example, we execute a single-line Windows command using the Execute VIX Script task, with the Output Display option disabled. When the Output Display option is unchecked, the Windows command is directly executed on the VM. A Batch file or an output file is not created.

Enter the following information in the Execute VIX Scripts task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. In the **Script** field, enter the following sample Windows command:

   ```
   C:\\WINDOWS\\system32\\cmd.exe /c mkdir C:\\test_Dir_1234
   ```
4. Uncheck **Output Display**, if it is checked.

After executing the task, you see the following information in the log:

```
Service Request ID: 458

Mar 21, 2016 10:03:00 UTC Request Submitted

Mar 21, 2016 10:03:03 UTC Executing workflow item number 1

Mar 21, 2016 10:03:03 UTC Completed workflow item number 0, with status completed

Mar 21, 2016 10:03:09 UTC Executing workflow item number 2

Mar 21, 2016 10:03:00 UTC Trigger context executeWorkflowStep called

Mar 21, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Mar 21, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Mar 21, 2016 10:03:09 UTC VIX API will be deprecated in future release.

Mar 21, 2016 10:03:13 UTC  [VIXActionHandler] - account details VMName : Win_2k12 , Host:
172.31.234.46, OS Type: microsoft windows server 2012 (64-bit), ExitCode :0

Mar 21, 2016 10:03:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 3 seconds

Mar 21, 2016 10:03:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Mar 21, 2016 10:03:13 UTC [Local Input: Select VM = 3312)

Mar 21, 2016 10:03:13 UTC [Local Input: Credential type = Login]

Mar 21, 2016 10:03:13 UTC [Local Input: Login =  administrator]

Mar 21, 2016 10:03:13 UTC [Local Input: Password = "masked-value")

Mar 21, 2016 10:03:13 UTC [Local Input: Script - C:\\WINDOWS\\system32\\cmd.exe /c mkdir
C:\\test_Dir_1234]

Mar 21, 2016 10:03:13 UTC [Local Input: Undo Script =  ]

Mar 21, 2016 10:03:13 UTC (Local Input: Error Codes = ]

Mar 21, 2016 10:03:13 UTC [Local Input: Output display = true]

Mar 21, 2016 10:03:13 UTC (Output: EXIT_STATUS_CODE = 0]

Mar 21, 2016 10:03:13 UTC (Output: ERROR_STATUS_MESSAGE = ]

Mar 21, 2016 10:03:13 UTC Completed workflow item number 1, with status completed

Mar 21, 2016 10:03:13 UTC Completed workflow item number 3

Mar 21, 2016 10:03:13 UTC Completed workflow item number 2, with status completed
```

As seen in the log, the command is executed directly on the VM.

Executing a Multiline Windows Command

In this example, we execute a multiline Windows command using the Execute VIX Script task, with the Output Display option disabled. When the Output Display option is unchecked, the multiline Windows command is directly executed on the VM. A Batch file or an output file is not created.

Enter the following information in the Execute VIX Scripts task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. In the **Script** field, enter the following sample Windows command:

   ```
   C:\\WINDOWS\\system32\\cmd.exe /c mkdir C:\\testfolder_line1  &&
   C:\\WINDOWS\\system32\\cmd.exe /c mkdir C:\\testfolder_line2
   ```

4. Uncheck **Output Display**, if it is checked.

After executing the task, you see the following information in the log:

```
Service Request ID: 448

Mar 21, 2016 10:03:00 UTC Request Submitted

Mar 21, 2016 10:03:03 UTC Executing workflow item number 1

Mar 21, 2016 10:03:03 UTC Completed workflow item number 0, with status completed

Mar 21, 2016 10:03:09 UTC Executing workflow item number 2

Mar 21, 2016 10:03:00 UTC Trigger context executeWorkflowStep called

Mar 21, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Mar 21, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Mar 21, 2016 10:03:09 UTC VIX API will be deprecated in future release.

Mar 21, 2016 10:03:13 UTC  [VIXActionHandler] - account details VMName : Win_2k12 , Host:
172.31.234.46, OS Type: microsoft windows server 2012 (64-bit), ExitCode :0

Mar 21, 2016 10:03:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 3 seconds

Mar 21, 2016 10:03:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Mar 21, 2016 10:03:13 UTC [Local Input: Select VM = 3312)

Mar 21, 2016 10:03:13 UTC [Local Input: Credential type = Login]

Mar 21, 2016 10:03:13 UTC [Local Input: Login =  administrator]

Mar 21, 2016 10:03:13 UTC [Local Input: Password = "masked-value")

Mar 21, 2016 10:03:13 UTC [Local Input: Script - C:\\WINDOWS\\system32\\cmd.exe /c mkdir
C:\\testfolder_line1  &&
C:\\WINDOWS\\system32\\cmd.exe /c mkdir C:\\testfolder_line2
]

Mar 21, 2016 10:03:13 UTC [Local Input: Undo Script =  ]

Mar 21, 2016 10:03:13 UTC (Local Input: Error Codes = ]

Mar 21, 2016 10:03:13 UTC [Local Input: Output display = true]

Mar 21, 2016 10:03:13 UTC (Output: EXIT_STATUS_CODE = 0]
```

```
Mar 21, 2016 10:03:13 UTC (Output: ERROR_STATUS_MESSAGE = ]

Mar 21, 2016 10:03:13 UTC Completed workflow item number 1, with status completed

Mar 21, 2016 10:03:13 UTC Completed workflow item number 3

Mar 21, 2016 10:03:13 UTC Completed workflow item number 2, with status completed
```

As seen in the log, the command is executed directly on the VM.

Executing a Single-Line Linux Command

In this example, we execute a single-line Linux command using the Execute VIX Script task, with the Output Display option disabled. When the Output Display option is disabled, the Linux command is directly executed on the VM. No output file is created.

Enter the following information in the Execute VIX Scripts task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. In the **Script** field, enter the following sample Linux command:
   `/bin/mkdir /tmp/testdir1`
4. Uncheck **Output Display**, if it is checked.

After executing the task, you see the following information in the SR log:

```
Apr 22, 2016 12:06:00 UTC Request Submitted

Apr 22, 2016 12:06:03 UTC Executing workflow item number 1

Apr 22, 2016 12:06:03 UTC Completed workflow item number 0, with status completed

Apr 22, 2016 12:06:09 UTC Executing workflow item number 2

Apr 22, 2016 12:06:00 UTC Trigger context executeWorkflowStep called

Apr 22, 2016 12:06:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 22, 2016 12:06:09 UTC Executing custom action Vix_script (Execute VIX Script)

Apr 22, 2016 12:06:09 UTC VIX API will be deprecated in future release.

Apr 22, 2016 12:06:13 UTC  [VIXActionHandler] - account details VMName : vm-DEF-SR119 ,
Host: 172.31.234.85, OS Type: red hat enterprise linux 4(32-bit), ExitCode :0

Apr 22, 2016 12:06:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 4 seconds

Apr 22, 2016 12:06:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 22, 2016 12:06:13 UTC [Local Input: Select VM = 9801)

Apr 22, 2016 12:06:13 UTC [Local Input: Credential type = Login]

Apr 22, 2016 12:06:13 UTC [Local Input: Login =  administrator]

Apr 22, 2016 12:06:13 UTC [Local Input: Password = "masked-value")

Apr 22, 2016 12:06:13 UTC [Local Input: Script - /bin/mkdir /tmp/testdir1]
```

```
Apr 22, 2016 12:06:13 UTC [Local Input: Undo Script =  ]
Apr 22, 2016 12:06:13 UTC (Local Input: Error Codes = ]
Apr 22, 2016 12:06:13 UTC [Local Input: Output display = true]
Apr 22, 2016 12:06:13 UTC (Output: EXIT_STATUS_CODE = 0]
Apr 22, 2016 12:06:13 UTC (Output: ERROR_STATUS_MESSAGE = ]
Apr 22, 2016 12:06:13 UTC Completed workflow item number 1, with status completed
```

As seen in the log, the Linux command is executed directly on the VM.

Examples with Guest Operations API Option Enabled

When the **Invoke Guest Operations API** option in the Execute VIX Scripts task is checked, the task uses the latest Guest OS access API layer instead of the VIX API. A batch file or an output file is not created.

The following examples illustrate how you can use the Execute VIX Script Task for running commands on Windows and Linux VMs, with the Guest Operations API option enabled.

Executing A Single-Line Windows Command

In this example, we execute a multiline Windows command using the Execute VIX Script task, with the Guest Operations API option enabled.

Enter the following information in the Execute VIX Scripts task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. In the **Script** field, enter the following sample Windows command:

   ```
   C:\\WINDOWS\\system32\\cmd.exe /c mkdir C:\\test_Dir_1234
   ```
4. Check **Invoke Guest Operations API**.

After executing the task, you see the following information in the log:

```
Service Request ID: 458
Mar 21, 2016 10:03:00 UTC Request Submitted
Mar 21, 2016 10:03:03 UTC Executing workflow item number 1
Mar 21, 2016 10:03:03 UTC Completed workflow item number 0, with status completed
Mar 21, 2016 10:03:09 UTC Executing workflow item number 2
Mar 21, 2016 10:03:00 UTC Trigger context executeWorkflowStep called
Mar 21, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)
Mar 21, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)
Mar 21, 2016 10:03:09 UTC Invoking Guest Operations API
Mar 21, 2016 10:03:13 UTC [VIXActionHandler] - Completed executing guest command
Mar 21, 2016 10:03:13 UTC Command invoked with Exit Code :0
```

```
Mar 21, 2016 10:03:13 UTC [VIXActionHandler] - account details VMName : Win_2k12 , Host:
172.31.234.46, OS Type: microsoft windows server 2012 (64-bit), ExitCode :0

Mar 21, 2016 10:03:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 3 seconds

Mar 21, 2016 10:03:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Mar 21, 2016 10:03:13 UTC [Local Input: Select VM = 3312)

Mar 21, 2016 10:03:13 UTC [Local Input: Credential type = Login]

Mar 21, 2016 10:03:13 UTC [Local Input: Login =  administrator]

Mar 21, 2016 10:03:13 UTC [Local Input: Password = "masked-value")

Mar 21, 2016 10:03:13 UTC [Local Input: Script - C:\\WINDOWS\\system32\\cmd.exe /c mkdir
C:\\test_Dir_1234]

Mar 21, 2016 10:03:13 UTC [Local Input: Undo Script =  ]

Mar 21, 2016 10:03:13 UTC (Local Input: Error Codes = ]

Mar 21, 2016 10:03:13 UTC [Local Input: Output display = true]

Mar 21, 2016 10:03:13 UTC (Output: EXIT_STATUS_CODE = 0]

Mar 21, 2016 10:03:13 UTC (Output: ERROR_STATUS_MESSAGE = ]

Mar 21, 2016 10:03:13 UTC Completed workflow item number 1, with status completed

Mar 21, 2016 10:03:13 UTC Completed workflow item number 3

Mar 21, 2016 10:03:13 UTC Completed workflow item number 2, with status completed
```

As seen in the log, the command is executed using the Guest Operations API.

Executing a Multiline Windows Command

In this example, we execute a multiline Windows command using the Execute VIX Script task, with the Guest Operations API option enabled.

Enter the following information in the Execute VIX Scripts task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. In the **Script** field, enter the following sample Windows command:

   ```
   C:\\WINDOWS\\system32\\cmd.exe /c mkdir C:\\testfolder_line1  &&
   C:\\WINDOWS\\system32\\cmd.exe /c mkdir C:\\testfolder_line2
   ```

4. Check **Invoke Guest Operations API**.

After executing the task, you see the following information in the log:

```
Service Request ID: 448

Mar 21, 2016 10:03:00 UTC Request Submitted

Mar 21, 2016 10:03:03 UTC Executing workflow item number 1

Mar 21, 2016 10:03:03 UTC Completed workflow item number 0, with status completed

Mar 21, 2016 10:03:09 UTC Executing workflow item number 2
```

```
Mar 21, 2016 10:03:00 UTC Trigger context executeWorkflowStep called

Mar 21, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Mar 21, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Mar 21, 2016 10:03:09 UTC Invoking Guest Operations API

Mar 21, 2016 10:03:13 UTC [VIXActionHandler] - Completed executing guest command

Mar 21, 2016 10:03:13 UTC Command invoked with Exit Code :0

Mar 21, 2016 10:03:13 UTC  [VIXActionHandler] - account details VMName : Win_2k12 , Host:
172.31.234.46, OS Type: microsoft windows server 2012 (64-bit), ExitCode :0

Mar 21, 2016 10:03:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 3 seconds

Mar 21, 2016 10:03:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Mar 21, 2016 10:03:13 UTC [Local Input: Select VM = 3312)

Mar 21, 2016 10:03:13 UTC [Local Input: Credential type = Login]

Mar 21, 2016 10:03:13 UTC [Local Input: Login =  administrator]

Mar 21, 2016 10:03:13 UTC [Local Input: Password = "masked-value")

Mar 21, 2016 10:03:13 UTC [Local Input: Script - C:\\WINDOWS\\system32\\cmd.exe /c mkdir
C:\\testfolder_line1  &&
C:\\WINDOWS\\system32\\cmd.exe /c mkdir C:\\testfolder_line2
]

Mar 21, 2016 10:03:13 UTC [Local Input: Undo Script =  ]

Mar 21, 2016 10:03:13 UTC (Local Input: Error Codes = ]

Mar 21, 2016 10:03:13 UTC [Local Input: Output display = true]

Mar 21, 2016 10:03:13 UTC (Output: EXIT_STATUS_CODE = 0]

Mar 21, 2016 10:03:13 UTC (Output: ERROR_STATUS_MESSAGE = ]

Mar 21, 2016 10:03:13 UTC Completed workflow item number 1, with status completed

Mar 21, 2016 10:03:13 UTC Completed workflow item number 3

Mar 21, 2016 10:03:13 UTC Completed workflow item number 2, with status completed
```

As seen in the log, the command is executed using the Guest Operations API.

Executing a Single-Line Linux Command

In this example, we execute a multiline Windows command using the Execute VIX Script task, with the Guest Operations API option enabled.

Enter the following information in the Execute VIX Scripts task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. In the **Script** field, enter the following sample Linux command:

   ```
   /bin/mkdir /tmp/testdir1
   ```

4. Check **Invoke Guest Operations API**.

After executing the task, you see the following information in the SR log:

```
Apr 22, 2016 12:06:00 UTC Request Submitted

Apr 22, 2016 12:06:03 UTC Executing workflow item number 1

Apr 22, 2016 12:06:03 UTC Completed workflow item number 0, with status completed

Apr 22, 2016 12:06:09 UTC Executing workflow item number 2

Apr 22, 2016 12:06:00 UTC Trigger context executeWorkflowStep called

Apr 22, 2016 12:06:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 22, 2016 12:06:09 UTC Executing custom action Vix_script (Execute VIX Script)

Apr 22, 2016 12:06:09 UTC Invoking Guest Operations API

Apr 22, 2016 12:06:13 UTC [VIXActionHandler] - Completed executing guest command

Apr 22, 2016 12:06:13 UTC Command invoked with Exit Code :0

Apr 22, 2016 12:06:13 UTC  [VIXActionHandler] - account details VMName : vm-DEF-SR119 ,
Host: 172.31.234.85, OS Type: red hat enterprise linux 4(32-bit), ExitCode :0

Apr 22, 2016 12:06:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 4 seconds

Apr 22, 2016 12:06:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 22, 2016 12:06:13 UTC [Local Input: Select VM = 9801)

Apr 22, 2016 12:06:13 UTC [Local Input: Credential type = Login]

Apr 22, 2016 12:06:13 UTC [Local Input: Login =  administrator]

Apr 22, 2016 12:06:13 UTC [Local Input: Password = "masked-value")

Apr 22, 2016 12:06:13 UTC [Local Input: Script - /bin/mkdir /tmp/testdir1]

Apr 22, 2016 12:06:13 UTC [Local Input: Undo Script =  ]

Apr 22, 2016 12:06:13 UTC (Local Input: Error Codes = ]

Apr 22, 2016 12:06:13 UTC [Local Input: Output display = true]

Apr 22, 2016 12:06:13 UTC (Output: EXIT_STATUS_CODE = 0]

Apr 22, 2016 12:06:13 UTC (Output: ERROR_STATUS_MESSAGE = ]

Apr 22, 2016 12:06:13 UTC Completed workflow item number 1, with status completed
```

As seen in the log, the command is executed using the Guest Operations API.

Executing VIX Scripts Using Rest APIs

The following examples illustrate how you can execute Linux or Windows commands on a VM using the following REST APIs:

- userAPIExecuteVIXScript
- EXECUTE_VIX_SCRIPT
- userAPIExecuteVIXScriptWithOutput

Executing a Windows Command Using the userAPIExecuteVIXScript API

In this example, we execute a Windows command using the userAPIExecuteVIXScript API task. The limitation of this API is that the response shows only the status code, and not the output message.

Enter the following information in the userAPIExecuteVIXScript API task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. Click Generate URL and enter the following parameters:

```
/app/api/rest?formatType=json&opName=genericvm:userAPIExecuteVIXScript&opData=
{param0:642,param1:"administrator",param2:"cloupia345",param3:"C:\\WINDOWS\\system32\\cmd.exe
 mkdir C:\\test_Dir"}
```

Note

- Here param0 is the VM ID, param1 and param2 provide the administrator credentials for the VM, and param3 is the Windows command.
- Use four backslashes in this API command if the command contains '\'. Verify the script in the service request (SR) logs.

4. Click **Execute REST API**.

If the command is valid, you see the following response:

```
{ "serviceResult":126, "serviceError":null, "serviceName":"InfraMgr",
"opName":"genericvm:userAPIExecuteVIXScript" }
```

If the command has been executed successfully, you see the following information in the SR log:

```
Service Request ID: 445

Apr 19, 2016 10:03:00 UTC Request Submitted

Apr 19, 2016 10:03:03 UTC Executing workflow item number 1

Apr 19, 2016 10:03:03 UTC Completed workflow item number 0, with status completed

Apr 19, 2016 10:03:09 UTC Executing workflow item number 2

Apr 19, 2016 10:03:00 UTC Trigger context executeWorkflowStep called

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC VIX API will be deprecated in future release.

Apr 19, 2016 10:03:13 UTC  [VIXActionHandler] - account details VMName : Win_2k12 , Host:
172.31.234.46, OS Type: microsoft windows server 2012 (64-bit), ExitCode :0

Apr 19, 2016 10:03:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 6 seconds

Apr 19, 2016 10:03:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 19, 2016 10:03:13 UTC [Local Input: Select VM = 642)

Apr 19, 2016 10:03:13 UTC [Local Input: Credential type = Login]
```

```
Apr 19, 2016 10:03:13 UTC [Local Input: Login =  administrator]

Apr 19, 2016 10:03:13 UTC [Local Input: Password = "masked-value")

Apr 19, 2016 10:03:13 UTC [Local Input: Script - C:\\WINDOWS\\system32\\cmd.exe mkdir
C:\\test_Dir]

Apr 19, 2016 10:03:13 UTC [Local Input: Undo Script =  ]

Apr 19, 2016 10:03:13 UTC (Local Input: Error Codes = ]

Apr 19, 2016 10:03:13 UTC [Local Input: Output display = true]

Apr 19, 2016 10:03:13 UTC (Output: EXIT_STATUS_CODE = 0]

Apr 19, 2016 10:03:13 UTC (Output: ERROR_STATUS_MESSAGE = ]

Apr 19, 2016 10:03:13 UTC Completed workflow item number 1, with status completed
```

As seen in the log, the script returns status code 0, which means that the task has been executed successfully.

If the command is invalid (for example you provide an incorrect command: C:\\WINDOWS\\system32\\cmd.exe testxml), the status code returns an appropriate error code in the SR log:

```
Service Request ID: 445

Apr 19, 2016 10:03:00 UTC Request Submitted

Apr 19, 2016 10:03:03 UTC Executing workflow item number 1

Apr 19, 2016 10:03:03 UTC Completed workflow item number 0, with status completed

Apr 19, 2016 10:03:09 UTC Executing workflow item number 2

Apr 19, 2016 10:03:00 UTC Trigger context executeWorkflowStep called

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC VIX API will be deprecated in future release.

Apr 19, 2016 10:03:13 UTC  [VIXActionHandler] - account details VMName : Win_2k12 , Host:
172.31.234.46, OS Type: microsoft windows server 2012 (64-bit), ExitCode :0

Apr 19, 2016 10:03:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
after 6 seconds

Apr 19, 2016 10:03:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 19, 2016 10:03:13 UTC [Local Input: Select VM = 642)

Apr 19, 2016 10:03:13 UTC [Local Input: Credential type = Login]

Apr 19, 2016 10:03:13 UTC [Local Input: Login =  administrator]

Apr 19, 2016 10:03:13 UTC [Local Input: Password = "masked-value")

Apr 19, 2016 10:03:13 UTC [Local Input: Script - C:\\WINDOWS\\system32\\cmd.exe testxml]

Apr 19, 2016 10:03:13 UTC [Local Input: Undo Script =  ]

Apr 19, 2016 10:03:13 UTC (Local Input: Error Codes = ]

Apr 19, 2016 10:03:13 UTC [Local Input: Output display = true]

Apr 19, 2016 10:03:13 UTC (Output: EXIT_STATUS_CODE = 9009]

Apr 19, 2016 10:03:13 UTC (Output: ERROR_STATUS_MESSAGE = 'testxml' is not recognized as
an internal or external command, operable program or batch file.]
```

```
Apr 19, 2016 10:03:13 UTC Completed workflow item number 1, with status completed
```

In this example, the status code 1 means an unknown error.

Executing a Linux Command Using the userAPIExecuteVIXScript API

In this example, we execute a Linux command using the userAPIExecuteVIXScript API task. The limitation of this API is that the response shows only the status code, and not the output message.

Enter the following information in the userAPIExecuteVIXScript API task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. Click **Generate URL** and enter the following parameters:

   ```
   /app/api/rest?formatType=json&opName=genericvm:userAPIExecuteVIXScript&opData=
   {param0:279,param1:"root",param2:"cloupia345",param3:"/bin/mkdir /temp/test1"}
   ```

Note Here param0 is the VM ID, param1 and param2 provide the root credentials for the VM, and param3 is the Linux command.

4. Click **Execute REST API**.

If the command is valid, you see the following response:

```
{ "serviceResult":81, "serviceError":null, "serviceName":"InfraMgr",
"opName":"genericvm:userAPIExecuteVIXScript" }
```

If the command has been executed successfully, you see the following information in the SR log:

```
Service Request ID: 495

Apr 19, 2016 10:03:00 UTC Request Submitted

Apr 19, 2016 10:03:03 UTC Executing workflow item number 1

Apr 19, 2016 10:03:03 UTC Completed workflow item number 0, with status completed

Apr 19, 2016 10:03:09 UTC Executing workflow item number 2

Apr 19, 2016 10:03:00 UTC Trigger context executeWorkflowStep called

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC VIX API will be deprecated in future release.

Apr 19, 2016 10:03:13 UTC  [VIXActionHandler] - account details VMName :SDK-VIX, Host
:172.29.110.75 , OS Type :red hat enterprise linux 5 (64-bit), ExitCode :0

Apr 19, 2016 10:03:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 6 seconds

Apr 19, 2016 10:03:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 19, 2016 10:03:13 UTC [Local Input: Select VM = 279)

Apr 19, 2016 10:03:13 UTC [Local Input: Credential type = Login]
```

```
Apr 19, 2016 10:03:13 UTC [Local Input: Login =  administrator]

Apr 19, 2016 10:03:13 UTC [Local Input: Password = "masked-value")

Apr 19, 2016 10:03:13 UTC [Local Input: Script - /bin/mkdir /temp/test1]

Apr 19, 2016 10:03:13 UTC [Local Input: Undo Script =  ]

Apr 19, 2016 10:03:13 UTC (Local Input: Error Codes = ]

Apr 19, 2016 10:03:13 UTC [Local Input: Output display = true]

Apr 19, 2016 10:03:13 UTC (Output: EXIT_STATUS_CODE = 0]

Apr 19, 2016 10:03:13 UTC (Output: ERROR_STATUS_MESSAGE = ]

Apr 19, 2016 10:03:13 UTC Completed workflow item number 1, with status completed
```

As seen in the log, the script returns status code 0, which means that the task has been executed successfully.

If the command is invalid (for example you provide an incorrect command: ///abcd/test/123), the status code returns an appropriate error code in the SR log:

```
Service Request ID: 495

Apr 19, 2016 10:03:00 UTC Request Submitted

Apr 19, 2016 10:03:03 UTC Executing workflow item number 1

Apr 19, 2016 10:03:03 UTC Completed workflow item number 0, with status completed

Apr 19, 2016 10:03:09 UTC Executing workflow item number 2

Apr 19, 2016 10:03:00 UTC Trigger context executeWorkflowStep called

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC VIX API will be deprecated in future release.

Apr 19, 2016 10:03:13 UTC  [VIXActionHandler] - account details VMName :SDK-VIX, Host
:172.29.110.75 , OS Type :red hat enterprise linux 5 (64-bit), ExitCode :0

Apr 19, 2016 10:03:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 3 seconds

Apr 19, 2016 10:03:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 19, 2016 10:03:13 UTC [Local Input: Select VM = 279)

Apr 19, 2016 10:03:13 UTC [Local Input: Credential type = Login]

Apr 19, 2016 10:03:13 UTC [Local Input: Login =  administrator]

Apr 19, 2016 10:03:13 UTC [Local Input: Password = "masked-value")

Apr 19, 2016 10:03:13 UTC [Local Input: Script - ///abcd/test/123]

Apr 19, 2016 10:03:13 UTC [Local Input: Undo Script =  ]

Apr 19, 2016 10:03:13 UTC (Local Input: Error Codes = ]

Apr 19, 2016 10:03:13 UTC [Local Input: Output display = true]

Apr 19, 2016 10:03:13 UTC (Output: EXIT_STATUS_CODE = 9009]

Apr 19, 2016 10:03:13 UTC (Output: ERROR_STATUS_MESSAGE = 'abcd/test/123' is not recognized
 as an internal or external command, operable program or batch file.]

Apr 19, 2016 10:03:13 UTC Completed workflow item number 1, with status completed
```

In this example, the status code 1 means an unknown error.

Executing a Windows Command Using the EXECUTE_VIX_SCRIPT API

In this example, we execute a Windows command using the EXECUTE_VIX_SCRIPT API task. The limitation of this API is that the response shows only the status code, and not the output message.

Enter the following information in the EXECUTE_VIX_SCRIPT API task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. In the **Script** field, enter the following sample Windows command:

   ```
   C:\\WINDOWS\\system32\\cmd.exe mkdir C:\\test_xml_api
   ```

4. Click **Execute REST API**.

If the command is valid, you see the following response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cuicOperationResponse>
    <Log>
         <Messages>1</Messages>
         <Message>
            <TimeStamp>2016-04-26 08:11:26.864</TimeStamp>
            <Severity>INFO</Severity>
           <Text>[VIXActionHandler] - account details VMName :SDK-VIX, Host :172.29.110.75
 , OS Type :microsoft windows server 2008 r2 (64-bit), ExitCode:0</Text>
         </Message>
   </Log>
   <operationStatus>0</operationStatus>
   <response>
       <ExecuteVIXScriptResponse>
           <EXIT_STATUS_CODE>0</EXIT_STATUS_CODE>
       </ExecuteVIXScriptResponse>
  </response>
  <responsemap>
     <entry>
       <key>ERROR_STATUS_CODE</key>
       <value>0</value>
     </entry>
  </responsemap>
</cuicOperationResponse>
```

As seen in the response, the status code returns 0, which means the command has been executed successfully.

If the command is invalid (for example you provide an incorrect command: C:\\WINDOWS\\system32\\cmd.exe testxml), the status code returns an appropriate error code in the Response field:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cuicOperationResponse>
    <Log>
         <Messages>1</Messages>
         <Message>
            <TimeStamp>2016-04-26 03:12:29.864</TimeStamp>
            <Severity>INFO</Severity>
           <Text>[VIXActionHandler] - account details VMName :SDK-VIX, Host :172.29.110.75
 , OS Type :red hat enterprise linux 5 (64-bit), ExitCode:0</Text>
         </Message>
    </Log>
    <operationStatus>0</operationStatus>
```

```
    <response>
       <ExecuteVIXScriptResponse>
           <EXIT_STATUS_CODE>9009</EXIT_STATUS_CODE>
           <ERROR_STATUS_MESSAGE>"testxml" is not recognized as an internal or external
command, operable program or batch file.</ERROR_STATUS_MESSAGE>
       </ExecuteVIXScriptResponse>
    </response>
    <responsemap>
      <entry>
        <key>ERROR_STATUS_CODE</key>
        <value>0</value>
      </entry>
      <entry>
        <key>ERROR_STATUS_MESSAGE</key>
        <value></value>
      </entry>
   </responsemap>
</cuicOperationResponse>
```

In this example, the status code 1 means an unknown error.

Executing a Linux Command Using the EXECUTE_VIX_SCRIPT API

In this example, we execute a Linux command using the EXECUTE_VIX_SCRIPT API task. The limitation of this API is that the response shows only the status code, and not the output message.

Enter the following information in the EXECUTE_VIX_SCRIPT API task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. In the Script field, enter the following sample Linux command:

   ```
   /bin/date
   ```

4. Click **Execute REST API**.

If the command is valid, you see the following response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cuicOperationResponse>
    <Log>
         <Messages>1</Messages>
         <Message>
            <TimeStamp>2016-04-26 03:11:26.864</TimeStamp>
            <Severity>INFO</Severity>
           <Text>[VIXActionHandler] - account details VMName :SDK-VIX, Host :172.29.110.75
 , OS Type :red hat enterprise linux 5 (64-bit), ExitCode:0</Text>
         </Message>
   </Log>
   <operationStatus>0</operationStatus>
   <response>
       <ExecuteVIXScriptResponse>
           <EXIT_STATUS_CODE>0</EXIT_STATUS_CODE>
           <ERROR_STATUS_MESSAGE></ERROR_STATUS_MESSAGE>
       </ExecuteVIXScriptResponse>
  </response>
  <responsemap>
     <entry>
       <key>ERROR_STATUS_CODE</key>
       <value>0</value>
     </entry>
    <entry>
      <key>ERROR_STATUS_MESSAGE</key>
```

```
      <value></value>
    </entry>
  </responsemap>
</cuicOperationResponse>
```

As seen in the response, the status code returns 0, which means the command has been executed successfully.

If the command is invalid, the status code returns an appropriate error code in the Response field:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cuicOperationResponse>
   <Log>
         <Messages>1</Messages>
         <Message>
            <TimeStamp>2016-04-26 03:11:26.864</TimeStamp>
            <Severity>INFO</Severity>
           <Text>[VIXActionHandler] - account details VMName :SDK-VIX, Host :172.29.110.75
 , OS Type :red hat enterprise linux 5 (64-bit), ExitCode:0</Text>
         </Message>
   </Log>
   <operationStatus>0</operationStatus>
   <response>
       <ExecuteVIXScriptResponse>
           <EXIT_STATUS_CODE>1</EXIT_STATUS_CODE>
           <ERROR_STATUS_MESSAGE></ERROR_STATUS_MESSAGE>
       </ExecuteVIXScriptResponse>
   </response>
   <responsemap>
      <entry>
        <key>ERROR_STATUS_CODE</key>
        <value>0</value>
        </entry>
     <entry>
        <key>ERROR_STATUS_MESSAGE</key>
        <value></value>
     </entry>
   </responsemap>
</cuicOperationResponse>
```

In this example, the status code 1 means an unknown error.

ExecutingaWindowsCommandUsingtheuserAPIExecuteVIXScriptWithOutput API

In this example, we execute a Windows command using the userAPIExecuteVIXScriptWithOutput API.

When you execute a Windows command using the userAPIExecuteVIXScriptWithOutput API, the API creates

- An output file (the file extension is .txt) in the file path location, /opt/infra/vix_temp_files.
- An Input/Output report that lists the input parameters and output variables for the workflow task.
- A batch file (the file extension is .bat) to verify the batch command (only for Windows VM).

Enter the following information in the userAPIExecuteVIXScriptWithOutput API task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.

3 Click **Generate URL** and enter the following parameters:

```
/app/api/rest?formatType=json&opName=genericvm:userAPIExecuteVIXScriptWithOutput&opData=
{param0:247,param1:"administrator",param2:"cloupia123",param3:"C:\\WINDOWS\\system32\\cmd.exe
 echo hi"}
```

Note

- Here param0 is the VM ID, param1 and param2 provide the administrator credentials for the VM, and param3 is the Windows command.
- For multiline commands, separate each line by \n. For example:

```
/app/api/rest?formatType=json&opName=genericvm:userAPIExecuteVIXScriptWithOutput&opData=
{param0:247,param1:"administrator",param2:"cloupia123",param3:"C:\\WINDOWS\\system32\\cmd.exe

echo hi\n echo this is secondline"}
```

4 Click **Execute REST API**.

If the command is valid, you see the following response:

```
{ "serviceResult":126, "serviceError":null, "serviceName":"InfraMgr",
"opName":"genericvm:userAPIExecuteVIXScriptWithOutput" }
```

If the command has been executed successfully, you see the following information in the SR log:

```
Service Request ID: 435

Apr 19, 2016 10:03:00 UTC Request Submitted

Apr 19, 2016 10:03:03 UTC Executing workflow item number 1

Apr 19, 2016 10:03:03 UTC Completed workflow item number 0, with status completed

Apr 19, 2016 10:03:09 UTC Executing workflow item number 2

Apr 19, 2016 10:03:00 UTC Trigger context executeWorkflowStep called

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC VIX API will be deprecated in future release.

Apr 19, 2016 10:03:13 UTC  [VIXActionHandler] - account details VMName : Win_2k12 , Host:
172.31.234.46, OS Type: microsoft windows server 2012 (64-bit), ExitCode :0

Apr 19, 2016 10:03:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 6 seconds

Apr 19, 2016 10:03:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 19, 2016 10:03:13 UTC [Local Input: Select VM = 247)

Apr 19, 2016 10:03:13 UTC [Local Input: Credential type = Login]

Apr 19, 2016 10:03:13 UTC [Local Input: Login =  administrator]

Apr 19, 2016 10:03:13 UTC [Local Input: Password = "masked-value")

Apr 19, 2016 10:03:13 UTC [Local Input: Script -
/app/api/rest?formatType=json&opName=genericvm:userAPIExecuteVIXScriptWithOutput&opData=
{param0:247,param1:"administrator",param2:"cloupia123",param3:"C:\\WINDOWS\\system32\\cmd.exe
 echo hi"}
]

Apr 19, 2016 10:03:13 UTC [Local Input: Undo Script =  ]

Apr 19, 2016 10:03:13 UTC (Local Input: Error Codes = ]
```

```
Apr 19, 2016 10:03:13 UTC [Local Input: Output display = true]

Apr 19, 2016 10:03:13 UTC (Output: EXIT_STATUS_CODE = 0]

Apr 19, 2016 10:03:13 UTC (Output: ERROR_STATUS_MESSAGE = ]

Apr 19, 2016 10:03:13 UTC Completed workflow item number 1, with status completed
```

As seen in the log, the script returns status code 0, which means that the task has been executed successfully.

The API also creates an Input/Output report that lists the input parameters and output variables (status code and status message) for the workflow task.

If an invalid Windows command is sent through the “userAPIExecuteVIXScriptWithOutput” API, you see the following report in the SR log:

```
Service Request ID: 435

Apr 19, 2016 10:03:00 UTC Request Submitted

Apr 19, 2016 10:03:03 UTC Executing workflow item number 1

Apr 19, 2016 10:03:03 UTC Completed workflow item number 0, with status completed

Apr 19, 2016 10:03:09 UTC Executing workflow item number 2

Apr 19, 2016 10:03:00 UTC Trigger context executeWorkflowStep called

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC VIX API will be deprecated in future release.

Apr 19, 2016 10:03:13 UTC  [VIXActionHandler] - account details VMName : Win_2k12 , Host:
172.31.234.46, OS Type: microsoft windows server 2012 (64-bit), ExitCode :0

Apr 19, 2016 10:03:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 9 seconds

Apr 19, 2016 10:03:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 19, 2016 10:03:13 UTC [Local Input: Select VM = 247)

Apr 19, 2016 10:03:13 UTC [Local Input: Credential type = Login]

Apr 19, 2016 10:03:13 UTC [Local Input: Login =  administrator]

Apr 19, 2016 10:03:13 UTC [Local Input: Password = "masked-value")

Apr 19, 2016 10:03:13 UTC [Local Input: Script -
/app/api/rest?formatType=json&opName=genericvm:userAPIExecuteVIXScriptWithOutput&opData=
{param0:247,param1:"administrator",param2:"cloupia123",param3:"C:\\WINDOWS\\system32\\cmd.exe
 whatsapp"}
]

Apr 19, 2016 10:03:13 UTC [Local Input: Undo Script =  ]

Apr 19, 2016 10:03:13 UTC (Local Input: Error Codes = ]

Apr 19, 2016 10:03:13 UTC [Local Input: Output display = true]

Apr 19, 2016 10:03:13 UTC (Output: EXIT_STATUS_CODE = 9009]

Apr 19, 2016 10:03:13 UTC (Output: ERROR_STATUS_MESSAGE = 'whatsapp' is not recognized as
an internal or external command, operable program or batch file.]

Apr 19, 2016 10:03:13 UTC Completed workflow item number 1, with status completed
```

As seen in the log, the script returns an appropriate status code and an error message that the command is invalid. The same error status code and error status message also appear in the Input/Output report.

The API creates a batch file in the following location:

```
/opt/infra/vix_temp_files/vixbatch-04_07_2016_02_01_34_231.bat
```

The API creates an output file in the following location:

```
/opt/infra/vix_temp_files/vixoutput-04_07_2016_02_01_34_231.txt
```

Executing a Linux Command Using the userAPIExecuteVIXScriptWithOutput API

In this example, we execute a Linux command using the userAPIExecuteVIXScriptWithOutput API.

Enter the following information in the userAPIExecuteVIXScriptWithOutput API task available on the UCS Director appliance:

1. Select the VM where you want to run the VIX script.
2. Enter the administrator credentials of the VM.
3. Click **Generate URL** and enter the following parameters:

   ```
   /app/api/rest?formatType=json&opName=genericvm:userAPIExecuteVIXScriptWithOutput&opData=
   {param0:244,param1:"root",param2:"cloupia123",param3:"/bin/date"}
   ```

Note Here param0 is the VM ID, param1 and param2 provide the administrator credentials for the VM, and param3 is the Linux command.

4. Click **Execute REST API**.

If the command is valid, you see the following response:

```
{ "serviceResult":228, "serviceError":null, "serviceName":"InfraMgr",
"opName":"genericvm:userAPIExecuteVIXScriptWithOutput" }
```

If the command has been executed successfully, you see the following information in the SR log:

```
Service Request ID: 491

Apr 19, 2016 10:03:00 UTC Request Submitted

Apr 19, 2016 10:03:03 UTC Executing workflow item number 1

Apr 19, 2016 10:03:03 UTC Completed workflow item number 0, with status completed

Apr 19, 2016 10:03:09 UTC Executing workflow item number 2

Apr 19, 2016 10:03:00 UTC Trigger context executeWorkflowStep called

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC VIX API will be deprecated in future release.

Apr 19, 2016 10:03:13 UTC  [VIXActionHandler] - account details VMName :SDK-VIX, Host
:172.29.110.75 , OS Type :red hat enterprise linux 5 (64-bit), ExitCode :0

Apr 19, 2016 10:03:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 6 seconds
```

```
Apr 19, 2016 10:03:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 19, 2016 10:03:13 UTC [Local Input: Select VM = 244)

Apr 19, 2016 10:03:13 UTC [Local Input: Credential type = Login]

Apr 19, 2016 10:03:13 UTC [Local Input: Login =  administrator]

Apr 19, 2016 10:03:13 UTC [Local Input: Password = "masked-value")

Apr 19, 2016 10:03:13 UTC [Local Input: Script - /bin/date}
]

Apr 19, 2016 10:03:13 UTC [Local Input: Undo Script =  ]

Apr 19, 2016 10:03:13 UTC (Local Input: Error Codes = ]

Apr 19, 2016 10:03:13 UTC [Local Input: Output display = true]

Apr 19, 2016 10:03:13 UTC (Output: EXIT_STATUS_CODE = 0]

Apr 19, 2016 10:03:13 UTC (Output: ERROR_STATUS_MESSAGE = ]

Apr 19, 2016 10:03:13 UTC Completed workflow item number 2, with status completed
```

As seen in the log, the script returns status code 0, which means that the task has been executed successfully.

The API also creates an Input/Output report that lists the input parameters and output variables (status code and status message) for the workflow task.

If an invalid Linux command (for example, "abcd") is sent through the “userAPIExecuteVIXScriptWithOutput” API, you see the following report in the SR log:

```
Service Request ID: 491

Apr 19, 2016 10:03:00 UTC Request Submitted

Apr 19, 2016 10:03:03 UTC Executing workflow item number 1

Apr 19, 2016 10:03:03 UTC Completed workflow item number 0, with status completed

Apr 19, 2016 10:03:09 UTC Executing workflow item number 2

Apr 19, 2016 10:03:00 UTC Trigger context executeWorkflowStep called

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC Executing custom action vix_script (Execute VIX Script)

Apr 19, 2016 10:03:09 UTC VIX API will be deprecated in future release.

Apr 19, 2016 10:03:13 UTC  [VIXActionHandler] - account details VMName :SDK-VIX, Host
:172.29.110.75 , OS Type :red hat enterprise linux 5 (64-bit), ExitCode :0

Apr 19, 2016 10:03:13 UTC Task#1 (vix_script (Execute VIX Script)) completed successfully
in 9 seconds

Apr 19, 2016 10:03:13 UTC Input/Output values for Task#1 (vix_script (Execute VIX Script)):

Apr 19, 2016 10:03:13 UTC [Local Input: Select VM = 244)

Apr 19, 2016 10:03:13 UTC [Local Input: Credential type = Login]

Apr 19, 2016 10:03:13 UTC [Local Input: Login =  administrator]

Apr 19, 2016 10:03:13 UTC [Local Input: Password = "masked-value")

Apr 19, 2016 10:03:13 UTC [Local Input: Script - abcd}

Apr 19, 2016 10:03:13 UTC [Local Input: Undo Script = null]
```

```
Apr 19, 2016 10:03:13 UTC (Output: EXIT_STATUS_CODE = 127]
Apr 19, 2016 10:03:13 UTC (Output: ERROR_STATUS_MESSAGE = sh:abcd: command not found]
Apr 19, 2016 10:03:13 UTC Completed workflow item number 2, with status completed
```

As seen in the log, the script returns an appropriate status code and an error message that the command is invalid. The same error status code and error status message also appear in the input/output report.

CHAPTER 4

Executing Commands through the Execute VM Command Task

This chapter contains the following sections:

Execute VM Command Task

You can also execute commands on guest VMs using the Execute VM Command task available from the UCS Director task library. Execute VM Command allows you to perform guest OS operations by running commands through either PowerShell or the command prompt.

To run PowerShell commands, provide the following in the Command Path field in the Execute VM Command task:

- The path of PowerShell
- The command along with the arguments

To run commands through the command prompt, provide the following in the Execute VM Command task:

- The path of cmd.exe in the Command Path field
- The command along with arguments in the Command Arguments field

Note The Execute VM Command task will be deprecated in a future release. The same functionality is available in the Guest Operations task.

See Guest Operations Task, on page 41.

Execute VM Command Task Examples

The following examples illustrate how you can execute Linux or Windows commands on a VM using the Execute VM Command task.

Executing a Windows Command at PowerShell

In this example, we execute a Windows command at PowerShell by using the Execute VM Command task. To execute a command at PowerShell, provide the path of PowerShell and the command.

Enter the following information in the Execute VM Command task available on the UCS Director appliance:

1. Select the VM where you want to execute the Windows command.
2. Enter the administrator credentials of the VM.
3. In the **Command Path** field, enter the path of PowerShell and the command:

```
C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\PowerShell.exe Get-NetAdapter |
Where-Object {$_.Status -eq 'UP'} | Set-DNSClientServerAddress -ServerAddresses
('${DNS1}','${DNS2}')
```

Note Here DNS1 and DNS2 are defined workflow inputs.

After executing the task, you see the following information in the SR log:

```
Service request ID: 734
Workflow Inputs:
- DNS1
- DNS2

Apr 26, 2016 12:06:00 UTC Request Submitted

Apr 26, 2016 12:06:03 UTC Executing workflow item number 1

Apr 26, 2016 12:06:03 UTC Completed workflow item number 0, with status completed

Apr 26, 2016 12:06:09 UTC Executing workflow item number 2

Apr 26, 2016 12:06:00 UTC Trigger context executeWorkflowStep called

Apr 26, 2016 12:06:09 UTC Executing custom action Execute_VM_Command(Execute VM Command)

Apr 26, 2016 12:06:09 UTC Executing custom action Execute_VM_Command(Execute VM Command)

Apr 26, 2016 12:06:13 UTC Task#1 (Execute_VM_Command(Execute VM Command)) completed
successfully in 3 seconds

Apr 26, 2016 12:06:13 UTC Input/Output values for Task#1 (Execute_VM_Command(Execute VM
Command)):

Apr 26, 2016 12:06:13 UTC [Local Input: Select VM = 3312)

Apr 26, 2016 12:06:13 UTC [Local Input: Login =  administrator]

Apr 26, 2016 12:06:13 UTC [Local Input: Password = "masked-value")

Apr 26, 2016 12:06:13 UTC [Template Input: Command Path =
C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\PowerShell.exe Get-NetAdapter | Where-Object
 {$_.Status -eq 'UP'} | Set-DNSClientServerAddress -ServerAddresses ('${DNS1}','${DNS2}')]
```

```
Apr 26, 2016 12:06:13 UTC [Resolved Template Input: Command Path =
C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\PowerShell.exe Get-NetAdapter | Where-Object
 {$_.Status -eq 'UP'} | Set-DNSClientServerAddress -ServerAddresses
('10.10.10.10','20.20.20.20')]

Apr 26, 2016 12:06:13 UTC [Local Input: Command Path =
C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\PowerShell.exe Get-NetAdapter | Where-Object
 {$_.Status -eq 'UP'} | Set-DNSClientServerAddress -ServerAddresses
('10.10.10.10','20.20.20.20')]


Apr 26, 2016 12:06:13 UTC [Local Input: Command Arguments = ]

Apr 26, 2016 12:06:13 UTC Completed workflow item number 1, with status completed

Apr 26, 2016 12:06:13 UTC Completed workflow item number 3

Apr 26, 2016 12:06:13 UTC Completed workflow item number 2, with status completed
```

Executing a Windows Command at Command Prompt

In this example, we execute a Windows command at Command prompt by using the Execute VM Command task. To execute a command at command prompt, provide the path of cmd.exe in the command path. Also provide the command and the comma-separated arguments in the Command Arguments field.

Enter the following information in the Execute VM Command task available on the UCS Director appliance:

1. Select the VM where you want to execute the Windows command.
2. Enter the administrator credentials of the VM.
3. In the **Command Path** field, enter the path of the `cmd.exe` file:

   `C:\\WINDOWS\\system32\\cmd.exe`

4. In the **Command Arguments** field, enter the command and the comma-separated arguments:

   `mkdir C:\\testDir1,C:\\testDir2,C:\\testDir3`

After executing the task, you see the following information in the SR log:

```
Service request ID: 534

Apr 06, 2016 12:06:00 UTC Request Submitted

Apr 06, 2016 12:06:03 UTC Executing workflow item number 1

Apr 06, 2016 12:06:03 UTC Completed workflow item number 0, with status completed

Apr 06, 2016 12:06:09 UTC Executing workflow item number 2

Apr 06, 2016 12:06:00 UTC Trigger context executeWorkflowStep called

Apr 06, 2016 12:06:09 UTC Executing custom action Execute_VM_Command(Execute VM Command)

Apr 06, 2016 12:06:09 UTC Executing custom action Execute_VM_Command(Execute VM Command)

Apr 06, 2016 12:06:13 UTC Task#1 (Execute_VM_Command(Execute VM Command)) completed
successfully in 3 seconds

Apr 06, 2016 12:06:13 UTC Input/Output values for Task#1 (Execute_VM_Command(Execute VM
Command)):

Apr 06, 2016 12:06:13 UTC [Local Input: Select VM = 3312)

Apr 06, 2016 12:06:13 UTC [Local Input: Login =  administrator]
```

```
Apr 06, 2016 12:06:13 UTC [Local Input: Password = "masked-value")

Apr 06, 2016 12:06:13 UTC [Local Input: Command Path = C:\\WINDOWS\\system32\\cmd.exe]

Apr 06, 2016 12:06:13 UTC [Local Input: Command Arguments = mkdir
C:\\testDir1,C:\\testDir2,C:\\testDir3]

Apr 06, 2016 12:06:13 UTC Completed workflow item number 1, with status completed

Apr 06, 2016 12:06:13 UTC Completed workflow item number 3

Apr 06, 2016 12:06:13 UTC Completed workflow item number 2, with status completed
```

Executing a Linux Command at Command Prompt

In this example, we execute a Linux command at Command prompt by using the Execute VM Command task.

Enter the following information in the Execute VM Command task available in the UCS Director appliance:

1. Select the VM where you want to run the Linux command.
2. Enter the administrator credentials of the VM.
3. In the **Command Path** field, enter the sample Linux command:

   `/bin/mkdir`

4. In the **Command Arguments** field, enter the command:

   `/opt/testDirectory`

After executing the task, you see the following information in the SR log:

```
Service request ID: 445

Apr 06, 2016 12:06:00 UTC Request Submitted

Apr 06, 2016 12:06:03 UTC Executing workflow item number 1

Apr 06, 2016 12:06:03 UTC Completed workflow item number 0, with status completed

Apr 06, 2016 12:06:09 UTC Executing workflow item number 2

Apr 06, 2016 12:06:00 UTC Trigger context executeWorkflowStep called

Apr 06, 2016 12:06:09 UTC Executing custom action Execute_VM_Command(Execute VM Command)

Apr 06, 2016 12:06:09 UTC Executing custom action Execute_VM_Command(Execute VM Command)

Apr 06, 2016 12:06:13 UTC Task#1 (Execute_VM_Command(Execute VM Command)) completed
successfully in 3 seconds

Apr 06, 2016 12:06:13 UTC Input/Output values for Task#1 (Execute_VM_Command(Execute VM
Command)):

Apr 06, 2016 12:06:13 UTC [Local Input: Select VM = 2339)

Apr 06, 2016 12:06:13 UTC [Local Input: Login =  administrator]

Apr 06, 2016 12:06:13 UTC [Local Input: Password = "masked-value")

Apr 06, 2016 12:06:13 UTC [Local Input: Command Path = /bin/mkdir]

Apr 06, 2016 12:06:13 UTC [Local Input: Command Arguments = mkdir /opt/testDirectory]

Apr 06, 2016 12:06:13 UTC Completed workflow item number 1, with status completed
```

```
Apr 06, 2016 12:06:13 UTC Completed workflow item number 3
Apr 06, 2016 12:06:13 UTC Completed workflow item number 2, with status completed
```

CHAPTER 5

Executing Commands through the Guest Operations Task

This chapter contains the following sections:

Guest Operations Task

You execute commands on guest VMs using the Guest Operations task available from the Cisco UCS Director task library. The Guest Operations task allows you to perform guest OS operations by running commands. The Guest Operations task supports rollback and the option to capture command output and error codes.

You can execute commands on Windows guest VMs through either PowerShell, a batch script, or the command prompt.

Note When you select a VM in the Guest Operations task, the OS type is automatically selected. You do not need to change the OS type.

The Guest Operations task has the same functionality as the existing Execute VIX Script task. In addition, the Guest Operations task supports:

- The option to upload a zip file to the Guest OS
- The option to upload and execute a zip file on the Guest OS

Note The uploaded zip file may contain one or more files to execute on the Guest OS.

The Guest Operations task extracts the zip file in a temporary location on Cisco UCS Director. For example, the files are extracted to `/opt/infra/uploads/permanent/admin/1492532488911` where *admin* is the system admin login name for Cisco UCS Director.

To run PowerShell commands, provide the following in the **Executable Path and Options** field in the Guest Operations task:

- The path of PowerShell.exe and options, such as `c:\Windows\System32\windowspowershell\v1.0\powershell.exe`

To run commands or batch script commands through the command prompt, provide the following in the Guest Operations task:

- The path of cmd.exe and options, such as `c:\Windows\System32\cmd.exe /c`
- The command in the Command Path field

Guest Operations Task Examples

The following examples illustrate how you can execute Linux or Windows commands on a VM using the Guest Operations task.

Executing a Single-Line Windows Command Using a Batch Script

In this example, we execute a single-line Windows command using the Guest Operations task.

Enter the following information in the Guest Operations task available on the Cisco UCS Director appliance:

1. Select the VM where you want to run the Windows batch script.
2. Enter the administrator credentials of the VM.
3. From the **OS Type** drop-down list, choose **Windows**.
4. In the **Executable Path and Options** field, enter the following sample Windows command:

   `C:\\WINDOWS\\system32\\cmd.exe /c`
5. In the **Command Path** field, enter the following sample command path:

   `mkdir C:\\test_Dir_1234`

After executing the task, you see the following information in the log:

```
Service Request ID: 1275

Jan 23, 2017 06:27:43 UTC Request submitted

Jan 23, 2017 06:27:43 UTC Executing workflow item number 1

Jan 23, 2017 06:27:43 UTC Completed workflow item number 1, with status Completed

Jan 23, 2017 06:27:47 UTC Executing workflow item number 2

Jan 23, 2017 06:27:47 UTC Trigger context executeWorkFlowStep called

Jan 23, 2017 06:27:47 UTC Executing custom action GO_Windows_Select_Batch (VMware Guest
Operations)

Jan 23, 2017 06:27:47 UTC Executing custom action GO_Windows_Select_Batch (VMware Guest
Operations)
```

```
Jan 23, 2017 06:27:47 UTC Started executing command on guest VIXTest_DND_2012

Jan 23, 2017 06:27:54 UTC Completed executing command

Jan 23, 2017 06:27:54 UTC Task #2 (GO_Windows_Select_Batch (VMware Guest Operations))
completed successfully in 6 seconds

Jan 23, 2017 06:27:54 UTC Input/Output values for Task #2 (GO_Windows_Select_Batch (VMware
 Guest Operations)):

Jan 23, 2017 06:27:54 UTC  [Local Input: Select VM = 16483]

Jan 23, 2017 06:27:54 UTC  [Local Input: User Name = Administrator]

Jan 23, 2017 06:27:54 UTC  [Local Input: Password = **masked-value**]

Jan 23, 2017 06:27:54 UTC  [Local Input: OS Type = windows]

Jan 23, 2017 06:27:54 UTC  [Local Input: Guest Operations = executeCommand]

Jan 23, 2017 06:27:54 UTC  [Local Input:  = null]

Jan 23, 2017 06:27:54 UTC  [Local Input: Guest File Path = null]

Jan 23, 2017 06:27:54 UTC  [Local Input: Executor Path and Arguments =
C:\\WINDOWS\\system32\\cmd.exe /c]

Jan 23, 2017 06:27:54 UTC  [Local Input: Command Path = mkdir C:\\test_Dir_1234]

Jan 23, 2017 06:27:54 UTC  [Local Input: Undo Script = ]

Jan 23, 2017 06:27:54 UTC  [Local Input: Error Codes = ]

Jan 23, 2017 06:27:54 UTC  [Output: EXIT_STATUS_CODE = 0]

Jan 23, 2017 06:27:54 UTC  [Output: COMMAND_OUTPUT = ]

Jan 23, 2017 06:27:54 UTC Completed workflow item number 2, with status Completed

Jan 23, 2017 06:27:59 UTC Executing workflow item number 3

Jan 23, 2017 06:27:59 UTC Completed workflow item number 3, with status Completed
```

As seen in the log, the command is executed directly on the VM.

Executing a Multiline Windows Command Using a Batch Script

In this example, we execute a multiline Windows command using the Guest Operations task.

Enter the following information in the Guest Operations task available on the Cisco UCS Director appliance:

1. Select the VM where you want to run the Windows batch script.
2. Enter the administrator credentials of the VM.
3. From the **OS Type** drop-down list, choose **Windows**.
4. In the **Executable Path and Options** field, enter the following sample Windows command:

   ```
   C:\\WINDOWS\\system32\\cmd.exe /c
   ```

5. In the **Command Path** field, enter the following sample command path:

   ```
   echo 1st file > C:\\One.txt && echo 2nd file > C:\\Two.txt
   ```

After executing the task, you see the following information in the log:

```
Service Request ID: 1328
```

```
Feb 01, 2017 06:15:17 UTC Request submitted

Feb 01, 2017 06:15:20 UTC Executing workflow item number 1

Feb 01, 2017 06:15:20 UTC Completed workflow item number 1, with status Completed

Feb 01, 2017 06:15:26 UTC Executing workflow item number 2

Feb 01, 2017 06:15:26 UTC Trigger context executeWorkFlowStep called

Feb 01, 2017 06:15:26 UTC Executing custom action Guest Operations testing (VMware Guest
Operations)

Feb 01, 2017 06:15:26 UTC Executing custom action Guest Operations testing (VMware Guest
Operations)

Feb 01, 2017 06:15:27 UTC Started executing command on guest VIXTest_DND_2012

Feb 01, 2017 06:15:33 UTC Completed executing command

Feb 01, 2017 06:15:33 UTC Task #2 (Guest Operations testing (VMware Guest Operations))
completed successfully in 6 seconds

Feb 01, 2017 06:15:33 UTC Input/Output values for Task #2 (Guest Operations testing (VMware
 Guest Operations)):

Feb 01, 2017 06:15:33 UTC  [Local Input: Select VM = 16483]

Feb 01, 2017 06:15:33 UTC  [Local Input: User Name = Administrator]

Feb 01, 2017 06:15:33 UTC  [Local Input: Password = **masked-value**]

Feb 01, 2017 06:15:33 UTC  [Local Input: OS Type = windows]

Feb 01, 2017 06:15:33 UTC  [Local Input: Guest Operations = executeCommand]

Feb 01, 2017 06:15:33 UTC  [Local Input: Upload File = null]

Feb 01, 2017 06:15:33 UTC  [Local Input: Guest File Path = null]

Feb 01, 2017 06:15:33 UTC  [Local Input: Executor Path and Options =
C:\Windows\System32\cmd.exe /c]

Feb 01, 2017 06:15:33 UTC  [Local Input: Command Path = echo 1st file > c:\\One.txt && echo
 2nd file > c:\\Two.txt]

Feb 01, 2017 06:15:33 UTC  [Local Input: Undo Script = ]

Feb 01, 2017 06:15:33 UTC  [Local Input: Error Codes = ]

Feb 01, 2017 06:15:33 UTC  [Output: EXIT_STATUS_CODE = 0]

Feb 01, 2017 06:15:33 UTC  [Output: COMMAND_OUTPUT = ]

Feb 01, 2017 06:15:33 UTC Completed workflow item number 2, with status Completed

Feb 01, 2017 06:15:38 UTC Executing workflow item number 3

Feb 01, 2017 06:15:38 UTC Completed workflow item number 3, with status Completed
```

As seen in the log, the command is executed directly on the VM.

Executing a Single-Line Linux Command

In this example, we execute a single-line Linux command using the Guest Operations task.

Enter the following information in the Guest Operations task available on the Cisco UCS Director appliance:

1 Select the VM where you want to run the Linux command.

2 Enter the administrator credentials of the VM.

3 From the **OS Type** drop-down list, choose **Linux**.

4 In the **Command Path** field, enter the following sample Linux command:

```
/bin/mkdir /tmp/testdir1
```

After executing the task, you see the following information in the log:

```
Service Request ID: 1272

Jan 23, 2017 06:17:45 UTC Request submitted

Jan 23, 2017 06:17:47 UTC Executing workflow item number 1

Jan 23, 2017 06:17:47 UTC Completed workflow item number 1, with status Completed

Jan 23, 2017 06:17:53 UTC Executing workflow item number 2

Jan 23, 2017 06:17:53 UTC Trigger context executeWorkFlowStep called

Jan 23, 2017 06:17:53 UTC Executing custom action GO_Linux_Select (VMware Guest Operations)

Jan 23, 2017 06:17:53 UTC Executing custom action GO_Linux_Select (VMware Guest Operations)

Jan 23, 2017 06:17:54 UTC Started executing command on guest VIXTest_DND_CentOS_Auto_prs

Jan 23, 2017 06:17:59 UTC Completed executing command

Jan 23, 2017 06:18:00 UTC Task #2 (GO_Linux_Select (VMware Guest Operations)) completed
successfully in 6 seconds

Jan 23, 2017 06:18:00 UTC Input/Output values for Task #2 (GO_Linux_Select (VMware Guest
Operations)):

Jan 23, 2017 06:18:00 UTC  [Local Input: Select VM = 16460]

Jan 23, 2017 06:18:00 UTC  [Local Input: User Name = root]

Jan 23, 2017 06:18:00 UTC  [Local Input: Password = **masked-value**]

Jan 23, 2017 06:18:00 UTC  [Local Input: OS Type = linux]

Jan 23, 2017 06:18:00 UTC  [Local Input: Guest Operations = executeCommand]

Jan 23, 2017 06:18:00 UTC  [Local Input:  = null]

Jan 23, 2017 06:18:00 UTC  [Local Input: Guest File Path = null]

Jan 23, 2017 06:18:00 UTC  [Local Input: Executor Path and Arguments = null]

Jan 23, 2017 06:18:00 UTC  [Local Input: Command Path = /bin/mkdir /tmp/testdir1]

Jan 23, 2017 06:18:00 UTC  [Local Input: Undo Script = /bin/rmdir /tmp/testdir1]

Jan 23, 2017 06:18:00 UTC  [Local Input: Error Codes = ]

Jan 23, 2017 06:18:00 UTC  [Output: EXIT_STATUS_CODE = 0]

Jan 23, 2017 06:18:00 UTC  [Output: COMMAND_OUTPUT = ]

Jan 23, 2017 06:18:00 UTC Completed workflow item number 2, with status Completed

Jan 23, 2017 06:18:02 UTC Executing workflow item number 3

Jan 23, 2017 06:18:02 UTC Completed workflow item number 3, with status Completed
```

As seen in the log, the command is executed directly on the VM.

Executing a Single-Line Windows Command at PowerShell

In this example, we execute a single-line Windows command at PowerShell by using the Guest Operations task. To execute a command at PowerShell, provide the path of PowerShell and the command.

Enter the following information in the Guest Operations task available on the Cisco UCS Director appliance:

1. Select the VM where you want to run the Windows command.
2. Enter the administrator credentials of the VM.
3. From the **OS Type** drop-down list, choose **Windows**.
4. In the **Executable Path and Options** field, enter the path of PowerShell:

   `C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\PowerShell.exe`
5. In the **Command Path** field, enter the following sample command path:

   `mkdir C:\\test_Dir_1234`

After executing the task, you see the following information in the log:

```
Service Request ID: 1277

Jan 23, 2017 06:32:27 UTC Request submitted

Jan 23, 2017 06:32:32 UTC Executing workflow item number 1

Jan 23, 2017 06:32:32 UTC Completed workflow item number 1, with status Completed

Jan 23, 2017 06:32:35 UTC Executing workflow item number 2

Jan 23, 2017 06:32:35 UTC Trigger context executeWorkFlowStep called

Jan 23, 2017 06:32:35 UTC Executing custom action GO_Windows_Select_PowerShell (VMware Guest
 Operations)

Jan 23, 2017 06:32:35 UTC Executing custom action GO_Windows_Select_PowerShell (VMware Guest
 Operations)

Jan 23, 2017 06:32:35 UTC Started executing command on guest VIXTest_DND_2012

Jan 23, 2017 06:32:41 UTC Completed executing command

Jan 23, 2017 06:32:42 UTC Task #2 (GO_Windows_Select_PowerShell (VMware Guest Operations))
 completed successfully in 6 seconds

Jan 23, 2017 06:32:42 UTC Input/Output values for Task #2 (GO_Windows_Select_PowerShell
(VMware Guest Operations)):

Jan 23, 2017 06:32:42 UTC  [Local Input: Select VM = 16483]

Jan 23, 2017 06:32:42 UTC  [Local Input: User Name = Administrator]

Jan 23, 2017 06:32:42 UTC  [Local Input: Password = **masked-value**]

Jan 23, 2017 06:32:42 UTC  [Local Input: OS Type = windows]

Jan 23, 2017 06:32:42 UTC  [Local Input: Guest Operations = executeCommand]

Jan 23, 2017 06:32:42 UTC  [Local Input:  = null]

Jan 23, 2017 06:32:42 UTC  [Local Input: Guest File Path = null]

Jan 23, 2017 06:32:42 UTC  [Local Input: Executor Path and Arguments =
C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\PowerShell.exe]

Jan 23, 2017 06:32:42 UTC  [Local Input: Command Path = mkdir C:\\test_Dir_1234]
```

```
Jan 23, 2017 06:32:42 UTC  [Local Input: Undo Script = ]

Jan 23, 2017 06:32:42 UTC  [Local Input: Error Codes = ]

Jan 23, 2017 06:32:42 UTC  [Output: COMMAND_OUTPUT = Directory: C:\Mode
LastWriteTime     Length Name

----               -------------     ------ ----          d----         1/22/2017  10:32
PM            test_Dir_1234]

Jan 23, 2017 06:32:42 UTC  [Output: EXIT_STATUS_CODE = 0]

Jan 23, 2017 06:32:42 UTC Completed workflow item number 2, with status Completed

Jan 23, 2017 06:32:44 UTC Executing workflow item number 3

Jan 23, 2017 06:32:44 UTC Completed workflow item number 3, with status Completed
```

As seen in the log, the command is executed directly on the VM.

Executing a Multiline Windows Command at PowerShell

In this example, we execute a multiline Windows command at PowerShell by using the Guest Operations task. To execute a command at PowerShell, provide the path of PowerShell and the command.

Enter the following information in the Guest Operations task available on the Cisco UCS Director appliance:

1. Select the VM where you want to run the Windows command.
2. Enter the administrator credentials of the VM.
3. From the **OS Type** drop-down list, choose **Windows**.
4. In the **Executable Path and Options** field, enter the path of PowerShell:

   ```
   C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\PowerShell.exe
   ```

5. In the **Command Path** field, enter the following sample command path:

   ```
   echo 1st file > C:\\One.txt
   echo 2nd file > C:\\Two.txt
   ```

After executing the task, you see the following information in the log:

```
Service Request ID: 1329

Feb 01, 2017 06:23:20 UTC Request submitted

Feb 01, 2017 06:23:21 UTC Executing workflow item number 1

Feb 01, 2017 06:23:21 UTC Completed workflow item number 1, with status Completed

Feb 01, 2017 06:23:24 UTC Executing workflow item number 2

Feb 01, 2017 06:23:24 UTC Trigger context executeWorkFlowStep called

Feb 01, 2017 06:23:24 UTC Executing custom action Guest Operations testing (VMware Guest
Operations)

Feb 01, 2017 06:23:24 UTC Executing custom action Guest Operations testing (VMware Guest
Operations)

Feb 01, 2017 06:23:25 UTC Started executing command on guest VIXTest_DND_2012

Feb 01, 2017 06:23:31 UTC Completed executing command

Feb 01, 2017 06:23:31 UTC Task #2 (Guest Operations testing (VMware Guest Operations))
completed successfully in 6 seconds
```

```
Feb 01, 2017 06:23:31 UTC Input/Output values for Task #2 (Guest Operations testing (VMware
 Guest Operations)):

Feb 01, 2017 06:23:31 UTC  [Local Input: Select VM = 16483]

Feb 01, 2017 06:23:31 UTC  [Local Input: User Name = Administrator]

Feb 01, 2017 06:23:31 UTC  [Local Input: Password = **masked-value**]

Feb 01, 2017 06:23:31 UTC  [Local Input: OS Type = windows]

Feb 01, 2017 06:23:31 UTC  [Local Input: Guest Operations = executeCommand]

Feb 01, 2017 06:23:31 UTC  [Local Input: Upload File = null]

Feb 01, 2017 06:23:31 UTC  [Local Input: Guest File Path = null]

Feb 01, 2017 06:23:31 UTC  [Local Input: Executor Path and Options =
C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\PowerShell.exe]

Feb 01, 2017 06:23:31 UTC  [Local Input: Command Path = echo 1st file > C:\\One.txt
echo 2nd file > C:\\Two.txt]

Feb 01, 2017 06:23:31 UTC  [Local Input: Undo Script = ]

Feb 01, 2017 06:23:31 UTC  [Local Input: Error Codes = ]

Feb 01, 2017 06:23:31 UTC  [Output: EXIT_STATUS_CODE = 0]

Feb 01, 2017 06:23:31 UTC Completed workflow item number 2, with status Completed

Feb 01, 2017 06:23:36 UTC Executing workflow item number 3

Feb 01, 2017 06:23:36 UTC Completed workflow item number 3, with status Completed
```

As seen in the log, the command is executed directly on the VM.

Uploading a File to a Guest VM

The Guest Operations task provides the option to upload a zipped batch script file or a zipped PowerShell script file to a Windows or Linux guest VM. If you select the Upload File to Guest option, you must choose the zip file and specify the path or directory on the guest VM to which you want to upload the zip file. The uploaded zip file may contain one or more files to execute on the Guest OS.

The Guest Operations task extracts the zip file in a temporary location on Cisco UCS Director. For example, the files are extracted to `/opt/infra/uploads/admin/1488281478583` where *admin* is the system admin login name for Cisco UCS Director.

Note To upload a zip file, you must have at least two times the file size available in Cisco UCS Director disk space.

Uploading a File to a Windows Guest VM

In this example, we upload a zip file to a Windows Guest VM by using the Guest Operations task.

Enter the following information in the Guest Operations task available on the Cisco UCS Director appliance:

1 Select the VM where you want to upload a zip file.

2 Enter the administrator credentials of the VM.

3 From the **OS Type** drop-down list, choose **Windows**.

4 From the **Guest Operations** drop-down list, choose **Upload File to Guest**.

5 In the **File** field, drag and drop the zip file you want to upload, or click **Select a File** to choose the zip file you want to upload.

6 In the **Guest OS Folder Path** field, enter the path on Windows guest VM to which you want to upload the zip file:

```
C:\FileUpload\
```

After executing the task, you see the following information in the log:

```
Service Request ID: 1282

Jan 23, 2017 07:01:28 UTC Request submitted

Jan 23, 2017 07:01:29 UTC Executing workflow item number 1

Jan 23, 2017 07:01:29 UTC Completed workflow item number 1, with status Completed

Jan 23, 2017 07:01:35 UTC Executing workflow item number 2

Jan 23, 2017 07:01:35 UTC Trigger context executeWorkFlowStep called

Jan 23, 2017 07:01:35 UTC Executing custom action GO_Windows_Upload (VMware Guest Operations)

Jan 23, 2017 07:01:35 UTC Executing custom action GO_Windows_Upload (VMware Guest Operations)

Jan 23, 2017 07:01:35 UTC Started uploading file to guest VIXTest_DND_2012

Jan 23, 2017 07:01:36 UTC Completed file uploading

Jan 23, 2017 07:01:36 UTC Task #2 (GO_Windows_Upload (VMware Guest Operations)) completed
successfully in 0 seconds

Jan 23, 2017 07:01:36 UTC Input/Output values for Task #2 (GO_Windows_Upload (VMware Guest
 Operations)):

Jan 23, 2017 07:01:36 UTC  [Local Input: Select VM = 16483]

Jan 23, 2017 07:01:36 UTC  [Local Input: User Name = Administrator]

Jan 23, 2017 07:01:36 UTC  [Local Input: Password = **masked-value**]

Jan 23, 2017 07:01:36 UTC  [Local Input: OS Type = windows]

Jan 23, 2017 07:01:36 UTC  [Local Input: Guest Operations = uploadFiletoGuest]

Jan 23, 2017 07:01:36 UTC  [Local Input:  = sample.zip]

Jan 23, 2017 07:01:36 UTC  [Local Input: Guest OS Folder Path = C:\FileUpload\]

Jan 23, 2017 07:01:36 UTC  [Local Input: Executor Path and Arguments = null]

Jan 23, 2017 07:01:36 UTC  [Local Input: Command Path = null]

Jan 23, 2017 07:01:36 UTC  [Local Input: Undo Script = ]

Jan 23, 2017 07:01:36 UTC  [Local Input: Error Codes = ]

Jan 23, 2017 07:01:36 UTC Completed workflow item number 2, with status Completed

Jan 23, 2017 07:01:38 UTC Executing workflow item number 3

Jan 23, 2017 07:01:38 UTC Completed workflow item number 3, with status Completed
```

As seen in the log, the command is executed directly on the VM.

Uploading a File to a Linux Guest VM

In this example, we upload a zip file to a Linux Guest VM by using the Guest Operations task.

Enter the following information in the Guest Operations task available on the Cisco UCS Director appliance:

1. Select the VM where you want to upload a zip file.
2. Enter the administrator credentials of the VM.
3. From the **OS Type** drop-down list, choose **Linux**.
4. From the **Guest Operations** drop-down list, choose **Upload File to Guest**.
5. In the **File** field, drag and drop the zip file you want to upload, or click **Select a File** to choose the zip file you want to upload.
6. In the **Guest OS Folder Path** field, enter the path on the Linux guest VM to which you want to upload the zip file:

   `/tmp/FileUpload/`

After executing the task, you see the following information in the log:

```
Service Request ID: 1280

Jan 23, 2017 06:44:57 UTC Request submitted

Jan 23, 2017 06:44:59 UTC Executing workflow item number 1

Jan 23, 2017 06:44:59 UTC Completed workflow item number 1, with status Completed

Jan 23, 2017 06:45:02 UTC Executing workflow item number 2

Jan 23, 2017 06:45:02 UTC Trigger context executeWorkFlowStep called

Jan 23, 2017 06:45:02 UTC Executing custom action GO_Linux_Upload (VMware Guest Operations)

Jan 23, 2017 06:45:02 UTC Executing custom action GO_Linux_Upload (VMware Guest Operations)

Jan 23, 2017 06:45:02 UTC Started uploading file to guest VIXTest_DND_CentOS

Jan 23, 2017 06:45:08 UTC Completed file uploading

Jan 23, 2017 06:45:08 UTC Task #2 (GO_Linux_Upload (VMware Guest Operations)) completed
successfully in 5 seconds

Jan 23, 2017 06:45:08 UTC Input/Output values for Task #2 (GO_Linux_Upload (VMware Guest
Operations)):

Jan 23, 2017 06:45:08 UTC  [Local Input: Select VM = 16485]

Jan 23, 2017 06:45:08 UTC  [Local Input: User Name = root]

Jan 23, 2017 06:45:08 UTC  [Local Input: Password = **masked-value**]

Jan 23, 2017 06:45:08 UTC  [Local Input: OS Type = linux]

Jan 23, 2017 06:45:08 UTC  [Local Input: Guest Operations = uploadFiletoGuest]

Jan 23, 2017 06:45:08 UTC  [Local Input:  = test.zip]

Jan 23, 2017 06:45:08 UTC  [Local Input: Guest OS Folder Path = /tmp/FileUpload/]

Jan 23, 2017 06:45:08 UTC  [Local Input: Executor Path and Arguments = null]

Jan 23, 2017 06:45:08 UTC  [Local Input: Command Path = null]
```

```
Jan 23, 2017 06:45:08 UTC  [Local Input: Undo Script = ]

Jan 23, 2017 06:45:08 UTC  [Local Input: Error Codes = ]

Jan 23, 2017 06:45:08 UTC Completed workflow item number 2, with status Completed

Jan 23, 2017 06:45:08 UTC Executing workflow item number 3

Jan 23, 2017 06:45:08 UTC Completed workflow item number 3, with status Completed
```

As seen in the log, the command is executed directly on the VM.

Executing an Uploaded File on a Guest VM

The Guest Operations task provides the option to execute an uploaded zipped batch script file or zipped PowerShell script file to a Windows or Linux guest VM. The uploaded zip file may contain one or more files to execute on the Guest OS.

Note To upload a zip file, you must have at least two times the file size available in Cisco UCS Director disk space.

The Guest Operations task extracts the zip file in a temporary location on Cisco UCS Director. For example, the files are extracted to `/opt/infra/uploads/`*`admin`*`/1488281478583` where *admin* is the system admin login name for Cisco UCS Director.

If you select the Execute Uploaded File option, you must specify the following:

- The file path on the guest VM to which you want to upload a single file that is zipped, or the directory on the guest VM to which you want to upload multiple files that are zipped
- The executor path and options (for Windows guest VMs)
- The command path

Executing an Uploaded Batch Script File on a Windows Guest VM

In this example, we upload and execute a batch script file on a Windows guest VM using the Guest Operations task.

Enter the following information in the Guest Operations task available on the Cisco UCS Director appliance:

1. Select the VM where you want to upload and exeute the Windows batch script.
2. Enter the administrator credentials of the VM.
3. From the **OS Type** drop-down list, choose **Windows**.
4. From the **Guest Operations** drop-down list, choose **Upload and Execute File**.
5. In the **File** field, drag and drop the zip file you want to upload, or click **Select a File** to choose the zip file you want to upload.
6. In the **Guest OS Folder Path** field, enter the path on Windows guest VM to which you want to upload the batch script file:

   `C:\FileUpload\`

7 In the **Executable Path and Options** field, enter the following sample Windows command:

```
C:\WINDOWS\system32\cmd.exe /c
```

8 In the **Command Path** field, enter the following sample command path:

```
CALL C:\FileUpload\Sample.bat
```

After executing the task, you see the following information in the log:

```
Service Request ID: 1294

Jan 23, 2017 07:50:20 UTC Request submitted

Jan 23, 2017 07:50:21 UTC Executing workflow item number 1

Jan 23, 2017 07:50:21 UTC Completed workflow item number 1, with status Completed

Jan 23, 2017 07:50:24 UTC Executing workflow item number 2

Jan 23, 2017 07:50:24 UTC Trigger context executeWorkFlowStep called

Jan 23, 2017 07:50:24 UTC Executing custom action GO_ExecuteUploadedFile_Batch (VMware Guest
 Operations)

Jan 23, 2017 07:50:24 UTC Executing custom action GO_ExecuteUploadedFile_Batch (VMware Guest
 Operations)

Jan 23, 2017 07:50:25 UTC Started uploading file to guest VIXTest_DND_2012

Jan 23, 2017 07:50:25 UTC Completed file uploading

Jan 23, 2017 07:50:25 UTC Started executing command on guest VIXTest_DND_2012

Jan 23, 2017 07:50:31 UTC Completed executing command

Jan 23, 2017 07:50:31 UTC Task #2 (GO_ExecuteUploadedFile_Batch (VMware Guest Operations))
 completed successfully in 6 seconds

Jan 23, 2017 07:50:31 UTC Input/Output values for Task #2 (GO_ExecuteUploadedFile_Batch
(VMware Guest Operations)):

Jan 23, 2017 07:50:31 UTC  [Local Input: Select VM = 16483]

Jan 23, 2017 07:50:31 UTC  [Local Input: User Name = Administrator]

Jan 23, 2017 07:50:31 UTC  [Local Input: Password = **masked-value**]

Jan 23, 2017 07:50:31 UTC  [Local Input: OS Type = windows]

Jan 23, 2017 07:50:31 UTC  [Local Input: Guest Operations = executeUploadedFile]

Jan 23, 2017 07:50:31 UTC  [Local Input:  = Sample.zip]

Jan 23, 2017 07:50:31 UTC  [Local Input: Guest OS Folder Path = C:\FileUpload\]

Jan 23, 2017 07:50:31 UTC  [Local Input: Executor Path and Arguments =
C:\Windows\System32\cmd.exe /c]

Jan 23, 2017 07:50:31 UTC  [Local Input: Command Path = CALL C:\FileUpload\Sample.bat]

Jan 23, 2017 07:50:31 UTC  [Local Input: Undo Script = null]

Jan 23, 2017 07:50:31 UTC  [Local Input: Error Codes = ]

Jan 23, 2017 07:50:31 UTC  [Output: EXIT_STATUS_CODE = 0]

Jan 23, 2017 07:50:31 UTC  [Output: COMMAND_OUTPUT = ]

Jan 23, 2017 07:50:31 UTC Completed workflow item number 2, with status Completed

Jan 23, 2017 07:50:36 UTC Executing workflow item number 3
```

```
Jan 23, 2017 07:50:36 UTC Completed workflow item number 3, with status Completed
```

As seen in the log, the command is executed directly on the VM.

Executing an Uploaded PowerShell Script File on a Windows Guest VM

In this example, we upload and execute a PowerShell script file on a Windows guest VM using the Guest Operations task. To execute a command at PowerShell, provide the path of PowerShell and the command.

Enter the following information in the Guest Operations task available on the Cisco UCS Director appliance:

1. Select the VM where you want to upload and execute the Windows PowerShell script.
2. Enter the administrator credentials of the VM.
3. From the **OS Type** drop-down list, choose **Windows**.
4. From the **Guest Operations** drop-down list, choose **Upload and Execute File**.
5. In the **File** field, drag and drop the zip file you want to upload, or click **Select a File** to choose the zip file you want to upload.
6. In the **Guest OS Folder Path** field, enter the path on Windows guest VM to which you want to upload the PowerShell script file:

   ```
   C:\FileUpload\
   ```
7. In the **Executable Path and Options** field, enter the following sample Windows command:

   ```
   C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\PowerShell.exe
   ```
8. In the **Command Path** field, enter the following sample command path:

   ```
   Invoke-Expression C:\FileUpload\sample.ps1
   ```

After executing the task, you see the following information in the log:

```
Service Request ID: 1295

Jan 23, 2017 07:56:51 UTC Request submitted

Jan 23, 2017 07:56:57 UTC Executing workflow item number 1

Jan 23, 2017 07:56:57 UTC Completed workflow item number 1, with status Completed

Jan 23, 2017 07:57:00 UTC Executing workflow item number 2

Jan 23, 2017 07:57:00 UTC Trigger context executeWorkFlowStep called

Jan 23, 2017 07:57:00 UTC Executing custom action GO_ExecuteUploadedFile_PowerShell (VMware
 Guest Operations)

Jan 23, 2017 07:57:00 UTC Executing custom action GO_ExecuteUploadedFile_PowerShell (VMware
 Guest Operations)

Jan 23, 2017 07:57:01 UTC Started uploading file to guest VIXTest_DND_2012

Jan 23, 2017 07:57:01 UTC Completed file uploading

Jan 23, 2017 07:57:01 UTC Started executing command on guest VIXTest_DND_2012

Jan 23, 2017 07:57:07 UTC Completed executing command

Jan 23, 2017 07:57:08 UTC Task #2 (GO_ExecuteUploadedFile_PowerShell (VMware Guest
Operations)) completed successfully in 7 seconds

Jan 23, 2017 07:57:08 UTC Input/Output values for Task #2 (GO_ExecuteUploadedFile_PowerShell
 (VMware Guest Operations)):
```

```
Jan 23, 2017 07:57:08 UTC  [Local Input: Select VM = 16483]

Jan 23, 2017 07:57:08 UTC  [Local Input: User Name = Administrator]

Jan 23, 2017 07:57:08 UTC  [Local Input: Password = **masked-value**]

Jan 23, 2017 07:57:08 UTC  [Local Input: OS Type = windows]

Jan 23, 2017 07:57:08 UTC  [Local Input: Guest Operations = executeUploadedFile]

Jan 23, 2017 07:57:08 UTC  [Local Input:  = sample.zip]

Jan 23, 2017 07:57:08 UTC  [Local Input: Guest OS Folder Path = C:\FileUpload\]

Jan 23, 2017 07:57:08 UTC  [Local Input: Executor Path and Arguments =
C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\PowerShell.exe]

Jan 23, 2017 07:57:08 UTC  [Local Input: Command Path = Invoke-Expression
C:\FileUpload\sample.ps1]

Jan 23, 2017 07:57:08 UTC  [Local Input: Undo Script = null]

Jan 23, 2017 07:57:08 UTC  [Local Input: Error Codes = ]

Jan 23, 2017 07:57:08 UTC  [Output: COMMAND_OUTPUT = Directory: C:\Mode
LastWriteTime     Length Name
----                -------------     ------ ----            d----         1/22/2017  11:56
 PM            testWinPSAPI]

Jan 23, 2017 07:57:08 UTC  [Output: EXIT_STATUS_CODE = 0]

Jan 23, 2017 07:57:08 UTC Completed workflow item number 2, with status Completed

Jan 23, 2017 07:57:09 UTC Executing workflow item number 3

Jan 23, 2017 07:57:09 UTC Completed workflow item number 3, with status Completed
```

As seen in the log, the command is executed directly on the VM.

Executing an Uploaded File on a Linux Guest VM

In this example, we upload and execute a file on a Linux guest VM using the Guest Operations task.

Enter the following information in the Guest Operations task available on the Cisco UCS Director appliance:

1. Select the VM where you want to run the upload and execute a Linux file.
2. Enter the administrator credentials of the VM.
3. From the **OS Type** drop-down list, choose **Linux**.
4. From the **Guest Operations** drop-down list, choose **Upload and Execute File**.
5. In the **File** field, drag and drop the zip file you want to upload, or click **Select a File** to choose the zip file you want to upload.
6. In the **Guest OS Folder Path** field, enter the path on Linux guest VM to which you want to upload the file:

   `/tmp/FileUpload/`
7. In the **Command Path** field, enter the following sample command path:

   `/bin/sh /tmp/FileUpload/test.sh`

After executing the task, you see the following information in the log:

```
Service Request ID: 1327
Feb 01, 2017 06:05:30 UTC Request submitted

Feb 01, 2017 06:05:32 UTC Executing workflow item number 1

Feb 01, 2017 06:05:32 UTC Completed workflow item number 1, with status Completed

Feb 01, 2017 06:05:38 UTC Executing workflow item number 2

Feb 01, 2017 06:05:38 UTC Trigger context executeWorkFlowStep called

Feb 01, 2017 06:05:38 UTC Executing custom action GO_Linux_ExecuteUploadedFile (VMware Guest
 Operations)

Feb 01, 2017 06:05:38 UTC Executing custom action GO_Linux_ExecuteUploadedFile (VMware Guest
 Operations)

Feb 01, 2017 06:05:38 UTC Started uploading file to guest VIXTest_DND_CentOS

Feb 01, 2017 06:05:44 UTC Completed file uploading

Feb 01, 2017 06:05:44 UTC Started executing command on guest VIXTest_DND_CentOS

Feb 01, 2017 06:05:50 UTC Completed executing command

Feb 01, 2017 06:05:50 UTC Task #2 (GO_Linux_ExecuteUploadedFile (VMware Guest Operations))
 completed successfully in 11 seconds

Feb 01, 2017 06:05:50 UTC Input/Output values for Task #2 (GO_Linux_ExecuteUploadedFile
(VMware Guest Operations)):

Feb 01, 2017 06:05:50 UTC  [Local Input: Select VM = 16485]

Feb 01, 2017 06:05:50 UTC  [Local Input: User Name = root]

Feb 01, 2017 06:05:50 UTC  [Local Input: Password = **masked-value**]

Feb 01, 2017 06:05:50 UTC  [Local Input: OS Type = linux]

Feb 01, 2017 06:05:50 UTC  [Local Input: Guest Operations = executeUploadedFile]

Feb 01, 2017 06:05:50 UTC  [Local Input: Upload File = test.zip]

Feb 01, 2017 06:05:50 UTC  [Local Input: Guest OS Folder Path = /tmp/FileUpload/]

Feb 01, 2017 06:05:50 UTC  [Local Input: Executor Path and Options = null]

Feb 01, 2017 06:05:50 UTC  [Local Input: Command Path = /bin/sh /tmp/FileUpload/test.sh]

Feb 01, 2017 06:05:50 UTC  [Local Input: Undo Script = null]

Feb 01, 2017 06:05:50 UTC  [Local Input: Error Codes = ]

Feb 01, 2017 06:05:50 UTC  [Output: EXIT_STATUS_CODE = 0]

Feb 01, 2017 06:05:50 UTC  [Output: COMMAND_OUTPUT = ]

Feb 01, 2017 06:05:50 UTC Completed workflow item number 2, with status Completed

Feb 01, 2017 06:05:50 UTC Executing workflow item number 3

Feb 01, 2017 06:05:50 UTC Completed workflow item number 3, with status Completed
```

As seen in the log, the command is executed directly on the VM.

CHAPTER 6

Troubleshooting

This chapter contains the following sections:

Debugging VIX Commands

Problem

Unable to execute VIX commands on a guest VM.

Solution

When you are unable to run a VIX command on a guest VM, you can run "cuicGuestCommand" at the UCS Director root shell. This command identifies whether the problem is because of lack of connectivity, an authentication error, or an incorrect command syntax.

Example

In the following example, we run the "cuicGuestCommand" at the UCSD root shell and identify that the problem is due to a file being not found.

```
[root@localhost vmsetup]# /opt/infra/inframgr/vmsetup/cuicGuestCommand http://192.0.2.1/sdk
 user903 pass103 "[TME_Demo_Prod_DS01] Dev_VM_SR344/Dev_VM_SR344.vmx" administrator pass1234
 C:\Windows\System32\cscript.exe
DEBUG: Success jobHandle  34603071
DEBUG: Success hostHandle 34603070
DEBUG: Before Opening VM
DEBUG: Opening VM....
DEBUG: Opened the VM
DEBUG: waiting for tools
DEBUG: tools up
DEBUG: logged in to guest
DEBUG: about to execute remote command
DEBUG: about to execute remote command C:WindowsSystem32cscript.exe
DEBUG: with args
failed to run program in virtual machine 4 ((null))
VMWARE_VIX_ERROR_CODE=4
VMWARE_VIX_ERROR_DESC=A file was not found
```

Escaping Spaces in a Windows Command

Problem

Unable to run a Windows command that contains spaces, when using the Execute VIX Script task.

Solution

To escape spaces in a command running on a Windows Guest VM, use the double quotes.

Example

Suppose that you want to add a domain group called "Sales Group" to a local administrators group in a Windows guest VM.

As the domain group contains a space (between Sales and Group), use double quotes. The syntaxes to run the command are as follows.

To run the command directly on the Windows Guest VM:

```
C:\Windows\System32\net.exe localgroup Administrators “Sales Group@domain.com” /ADD
```

To run the command through the Execute VIX Script Task (without ${Variable}):

```
C:\\Windows\\System32\\net.exe localgroup Administrators  \"“Sales Group@domain.com\"”  /ADD
```

To run the command through the Execute VIX Script Task (with ${Variable}):

```
C:\\\\Windows\\\\System32\\\\net.exe localgroup Administrators  \\”“${DomainGroup}\\"”  /ADD
```