Using a Proxy with vSphere Virtual Serial Ports

vSphere Web Services SDK 5.1

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/support/pubs.
# Contents

About This Book  5

1 Remote Serial Port Access with a Virtual Serial Port Proxy  7
   Virtual Serial Port Proxy Capabilities  8  
   Proxy as a Virtual Serial Port Concentrator  8  
   Network Serial Ports  9  
   Backing Options and Backing Information for Network Serial Ports  9  
   Network Connections Between a Virtual Machine, a Proxy, and a Remote System  10  
   Determining Backing Options  11  
   Creating a Virtual Machine with a Network Serial Port  11  
   VMware Telnet Session  12  
   Establishing a VMware Telnet Extension Session  13  
   vMotion and Persistent Serial Port Connections  13

2 VMware Telnet Extension Commands for Proxy Communication  17  
   Command Syntax  17  
   VMware Telnet Extension Option  18  
   Option Subnegotiation  18  
   Unknown Command Response  19  
   Proxy Operation  20  
   vMotion Notification  20  
   Virtual Machine Identification  22

Index  23
About This Book

The VMware® Using a Proxy with vSphere Virtual Serial Ports manual provides information about developing a virtual serial port concentrator for use in the vSphere environment.

VMware provides different APIs and SDKs for various applications and goals. This manual describes the capabilities of a proxy implementation that supports the VMware telnet extension. It also includes information about configuring a virtual serial port for a proxy connection.

To view the current version of this book as well as all VMware API and SDK documentation, go to http://www.vmware.com/support/pubs/sdk_pubs.html.

Revision History

This book is revised with each release of the product or when necessary. A revised version can contain minor or major changes. Table 1 summarizes the significant changes in each version of this book.

Table 1. Revision History

<table>
<thead>
<tr>
<th>Revision Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10Sep2012</td>
<td>Supported in VMware vSphere 5.1; added syntax for authentication parameters used with encrypted telnet and telnets connections.</td>
</tr>
<tr>
<td>24Aug2011</td>
<td>Supported in VMware vSphere 5.0.</td>
</tr>
<tr>
<td>13JUL2010</td>
<td>Initial release, supported in VMware vSphere 4.1.</td>
</tr>
</tbody>
</table>

Intended Audience

This book is intended for anyone developing a virtual serial port proxy for the vSphere environment. Proxy developers should have experience with the telnet protocol and network communication.

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation go to http://www.vmware.com/support/pubs.

Document Feedback

VMware welcomes your suggestions for improving our documentation. Send your feedback to docfeedback@vmware.com.
Remote Serial Port Access with a Virtual Serial Port Proxy

The VMware vSphere™ API supports the use of virtual serial ports on virtual machines that run on ESX™ hosts. A virtual serial port represents a serial port on a virtual machine. A virtual serial port can operate as a network serial port by using a network socket on the host to provide access to remote systems. A remote system is any system on the network that supports a serial port connection with a virtual machine. This feature supports remote console login for system management using an out-of-band connection that bypasses the vSphere Client.

You can also use a virtual serial port for communication through a software pipe, for output to a file, and for communication through a physical serial port on the host. For information about the vSphere API support for these applications, see the description of the VirtualSerialPort object in the vSphere API Reference.

In a VMware® datacenter, a vCenter Server uses vMotion™ to move virtual machines across hosts as part of resource management. vSphere does not support vMotion operations on a virtual machine that has a direct network serial port connection with a remote system. Network serial ports use the host IP address. If the Server attempts to move a virtual machine with a network serial port connection, the host rejects the vMotion request to maintain the connection.

To support vMotion for virtual machines that use network serial ports, you can develop a proxy that operates between a virtual machine and a remote system. The vSphere API supports the use of a third-party virtual serial port concentrator in the datacenter environment. A virtual serial port concentrator acts as a proxy or an access server between virtual machines and remote systems. The proxy maintains communication with virtual machines that are using network serial ports. When a vMotion event occurs, the virtual machine connects to the proxy from its new host so that the connection with the remote system persists.

Virtual machines use the telnet protocol for virtual serial port proxy connections. VMware defines a telnet extension that includes commands for proxy notification of vMotion events.

This chapter provides the following information:
- “Virtual Serial Port Proxy Capabilities” on page 8
- “Network Serial Ports” on page 9
- “VMware Telnet Session” on page 12
- “vMotion and Persistent Serial Port Connections” on page 13

Virtual Serial Port Proxy Capabilities

A virtual serial port proxy operates as an RFC2217-compatible telnet server. A proxy provides the following capabilities in the vSphere environment.

- Accepts telnet connections from virtual machines
- Supports VMware telnet extension commands and responds to VMware telnet extension messages from virtual machines
- Forwards network traffic between virtual machines and remote systems
- Handles extension commands for notification of vMotion events and redirects traffic accordingly

Proxy as a Virtual Serial Port Concentrator

Figure 1-1 shows a proxy that operates as a virtual serial port concentrator. The concentrator has internal connections to several virtual machines in a datacenter and an external connection to a remote system.

The concentrator provides a remote system with access to multiple virtual machines that act as serial port servers. The virtual machines connect to the concentrator, indicating that they will accept connections from remote systems. See “Backing Options and Backing Information for Network Serial Ports” on page 9 for more information about serial port servers. The concentrator presents the choice of virtual machines to the remote system and maintains any subsequent connection. In Figure 1-1, a solid line represents an active connection between the remote system and a virtual machine, and dashed lines represent potential connections.

Figure 1-1. Virtual Serial Port Concentrator

Figure 1-1 shows the concentrator as part of the vSphere datacenter. For this kind of application, you can deploy the concentrator as a virtual appliance. A virtual appliance is software on a platform of one or more virtual machines (in this case, a single virtual machine). A virtual appliance implementation allows you to take advantage of resource management that vCenter Server provides. The vCenter server uses vMotion technology to migrate the virtual appliance proxy through the datacenter without interrupting remote serial port service. The virtual appliance uses virtual machine IP addresses that persist across vMotion events.

Although the primary purpose of the proxy is to preserve connections during vMotion events, it also can provide a gateway or firewall to enforce security. The proxy serves as the point of authentication for remote systems that use virtual serial ports to access virtual machines.
Network Serial Ports

A network serial port is a virtual serial port that uses a network socket on the host computer. A network socket is the endpoint of a network connection and is represented by an IP address and a port number. The vSphere API uses Uniform Resource Identifiers (URI) to identify the protocol, IP address, and port number for endpoints of network serial port connections. See RFC3986 for information about Uniform Resource Identifiers.

To use a network serial port, you must verify that the ESX host supports the feature and then create or reconfigure a virtual machine to configure the port. For information about writing an application that uses the vSphere API, see the vSphere Web Services SDK Programming Guide. As an alternative, you can use the vSphere Client to configure a network serial port. For more information, see the online help for the VMware vSphere Client.

Backing Options and Backing Information for Network Serial Ports

The vSphere API uses backing objects to represent the host device that is associated with a virtual device. The types of backing objects for virtual devices are option objects and information objects.

Backing option objects identify host device support for virtual machines. An ESX host defines the set of virtual device backing options that are available for virtual machines. To use a network serial port, the ESX host must provide the virtual serial port URI backing option, which is represented by the VirtualSerialPortURIBackingOption object. For information about how to find out if the host supports the URI backing option, see “Determining Backing Options” on page 11.

Backing information objects supply data for virtual device configuration. Use a VirtualSerialPortURIBackingInfo object to bind a virtual serial port to a network socket. You can configure a virtual serial port to use URI backing when you create or reconfigure a virtual machine. For a description of how to configure URI backing for a network serial port, see “Creating a Virtual Machine with a Network Serial Port” on page 11.

When you specify URI backing for the network serial port connection, you define these characteristics.

- Direction – defines the role that the virtual machine plays in a connection with a remote system (client or server). When a virtual machine acts as a serial port client, it initiates the connection with the remote system. When a virtual machine acts as a serial port server, it listens for connections that are initiated by remote clients.

- Service URI – identifies the target of connections. When the virtual machine acts as a client, the service URI identifies the remote system. When the virtual machine acts as a server, the service URI identifies proxy access to the host on which the virtual machine runs. See RFC3986 for information about URI syntax.

- Proxy URI – identifies the virtual serial port proxy. The presence of the proxy URI directs the virtual machine to establish a connection with the proxy. The virtual machine always initiates the connection with the proxy.

You can use either telnet or secure telnet (telnets) for the proxy connection. The proxy URI specification must use one of the following formats. See RFC3986 for information about URI syntax. See the description of the VirtualSerialPortURIBackingInfo data object in the vSphere API Reference for information about URI schemes and authentication parameters for network serial port configuration.

- telnet://<host>:<port>[#key=[value][&key=[value] ...]]

- telnets://<host>:<port>[#key=[value][&key=[value] ...]]

When the virtual machine performs telnet negotiation with the proxy, it sends the direction and service URI information to the proxy. (See the description of the D0–PROXY command in “Proxy Operation” on page 20.)
Network Connections Between a Virtual Machine, a Proxy, and a Remote System

Figure 1-2 shows two virtual machines. Each machine uses a proxy for its connection to a remote system.

- One virtual machine acts as a virtual serial port server. This virtual machine uses the service URI to identify proxy access to the host machine. The virtual machine initiates a telnet connection with the proxy. The proxy, on behalf of the virtual machine, listens for connection requests from remote systems and forwards the requests to the virtual machine.

- One virtual machine acts as a virtual serial port client. This virtual machine uses the service URI to identify the remote system. The virtual machine initiates a telnet connection with the proxy. The proxy forwards the connection request to the remote system.

The proxy has two network sockets. It uses one socket (proxy URI) for the telnet connections with the virtual machine and the other for the connection with the remote system. The external connection between the proxy and a remote system uses the mutually agreed upon protocol. For a connection initiated by the remote system, the person operating the remote system must obtain the service URI by email or other independent communication.

**Figure 1-2. Virtual Serial Port Connections to a Remote System Through a Proxy**

Virtual machine as virtual serial port server

Virtual machine as virtual serial port client

---

**Figure 1-2 Diagram Details**

- ESX Server
- Virtual machine
- Virtual serial port
- Network socket
- IP address: port number
- Telnet
- Proxy URI
- Virtual serial port proxy
- Service URI
- IP address: port number
- Remote system
- Connection request
Determining Backing Options

You use the QueryConfigOption method to determine the backing options that are available on the host on which you will create the virtual machine. The method returns a VirtualMachineConfigOption data object. The virtual machine configuration data includes a list of virtual device options and a list of backing options. The following pseudocode shows the path to the backing options.

VirtualMachineConfigOption.hardwareOptions.VirtualDeviceOption[].backingOption[]

The virtual device option array must include the VirtualSerialPortOption data object. For network backing, the array of serial port backing options must include the VirtualDeviceURIBackingOption data object. The VirtualDeviceURIBackingOption.type property specifies the name of the backing object you should create when you configure network backing.

Creating a Virtual Machine with a Network Serial Port

You can use the CreateVM_Task method to create a virtual machine and configure a network serial port. The virtual machine uses the virtual serial port configuration to connect to a proxy and to accept a connection from a remote system.

1. Create a VirtualMachineConfigSpec object for the method's config parameter.
2. Create a VirtualDeviceConfigSpec object for the deviceChange array in the VirtualMachineConfigSpec object.
3. Create a VirtualSerialPort object for the device property in the VirtualDeviceConfigSpec object.
4. To control proxy connection behavior, create a VirtualDeviceConnectInfo data object for the VirtualSerialPort.connectable property.

The following pseudocode shows the resulting path to the connection information.

VirtualMachineConfigSpec.deviceChange[].device.connectable

5. Set the startConnected property in the connectable object to TRUE.

The startConnected property determines whether the virtual machine will open a connection with the proxy when the virtual machine starts. When the virtual machine powers on, it copies the startConnected value to the VirtualDeviceConnectInfo.connected property. While the virtual machine is running, you can use the ReconfigVM_Task method to set the connected property to open or close the proxy connection. If you reconfigure a port with connected set to TRUE, the virtual machine closes the existing connection and opens a new one.

6. Create a VirtualDeviceURIBackingInfo object for the backing property in the VirtualSerialPort object.

The following pseudocode shows the resulting path to the backing information.

VirtualMachineConfigSpec.deviceChange[].device.backing

Use the backing information object to supply the following information:

- Set the direction property to “server” to direct the virtual machine to accept remote connections.
- Set the serviceURI property to the URI for the host on which the virtual machine runs.
- Set the proxyURI property to the URI for the virtual serial port concentrator. You cannot include a user/password sequence in the proxy URI.
VMware Telnet Session

To establish a VMware telnet session for remote serial port communication, a virtual machine and a proxy participate in a message exchange that follows the model of a standard telnet protocol sequence. The proxy handles messages from the virtual machine and sends the appropriate responses. Before the proxy can forward content, the virtual machine and the proxy participate in the following progression.

1. The virtual machine and the proxy negotiate the use of the VMware telnet extension option.
   Both sides of the connection must agree to use the option before continuing with any additional extension message exchange.

2. The virtual machine and the proxy identify the VMware telnet suboption commands that they support.
   The proxy must support the KNOWN-SUBOPTIONS-2 and UNKNOWN-SUBOPTION-RCVD-2 commands. During a session, the proxy can send the UNKNOWN-SUBOPTION-RCVD-2 command as a valid response to virtual machine messages.

3. The proxy agrees to act as a VMware telnet extension proxy for remote serial port communication.

Table 1-1 lists the VMware telnet extension commands along with brief descriptions. Commands in the table are grouped by purpose: VMware telnet extension and suboptions, proxy, vMotion, and identification. The table uses command symbols. A VMware telnet extension command is a two- or three-byte code embedded in a telnet command sequence. For more information about commands and the code equivalents, see Chapter 2, “VMware Telnet Extension Commands for Proxy Communication,” on page 17.

Table 1-1. VMware Telnet Extension Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMAIRE-TELNET-EXT</td>
<td>virtual machine, proxy</td>
<td>The VMware telnet extension option command</td>
</tr>
<tr>
<td>KNOWN-SUBOPTIONS-1</td>
<td>virtual machine</td>
<td>Suboptions supported by the virtual machine</td>
</tr>
<tr>
<td>KNOWN-SUBOPTIONS-2</td>
<td>proxy</td>
<td>Suboptions supported by the proxy</td>
</tr>
<tr>
<td>UNKNOWN-SUBOPTION-RCVD-1</td>
<td>virtual machine</td>
<td>Unknown suboption received by the virtual machine</td>
</tr>
<tr>
<td>UNKNOWN-SUBOPTION-RCVD-2</td>
<td>proxy</td>
<td>Unknown suboption received by the proxy</td>
</tr>
<tr>
<td>DO-PROXY</td>
<td>virtual machine</td>
<td>Virtual machine request for proxy support</td>
</tr>
<tr>
<td>WILL-PROXY</td>
<td>proxy</td>
<td>Positive proxy response</td>
</tr>
<tr>
<td>WON’T-PROXY</td>
<td>proxy</td>
<td>Negative proxy response</td>
</tr>
<tr>
<td>VMOTION-BEGIN</td>
<td>virtual machine</td>
<td>Notification of a pending vMotion operation for a virtual machine</td>
</tr>
<tr>
<td>VMOTION-GOHEAD</td>
<td>proxy</td>
<td>Positive proxy response</td>
</tr>
<tr>
<td>VMOTION-NOTNOW</td>
<td>proxy</td>
<td>Negative proxy response</td>
</tr>
<tr>
<td>VMOTION-PEER</td>
<td>virtual machine</td>
<td>Virtual machine request identifying the new virtual machine instance</td>
</tr>
<tr>
<td>VMOTION-PEER-OK</td>
<td>proxy</td>
<td>Proxy response accepting the new virtual machine instance</td>
</tr>
<tr>
<td>VMOTION-COMPLETE</td>
<td>virtual machine</td>
<td>Notification that vMotion operation is complete</td>
</tr>
<tr>
<td>VMOTION-ABORT</td>
<td>virtual machine</td>
<td>Notification that vMotion operation was aborted</td>
</tr>
<tr>
<td>GET-VM-VC-UUID</td>
<td>proxy</td>
<td>Proxy request for the virtual center representation of virtual machine UUID</td>
</tr>
<tr>
<td>VM-VC-UUID</td>
<td>virtual machine</td>
<td>Virtual machine response</td>
</tr>
<tr>
<td>GET-VM-NAME</td>
<td>proxy</td>
<td>Proxy request for the virtual machine name</td>
</tr>
<tr>
<td>VM-NAME</td>
<td>virtual machine</td>
<td>Virtual machine response</td>
</tr>
<tr>
<td>GET-VM-BIOS-UUID</td>
<td>proxy</td>
<td>Proxy request for the virtual machine BIOS UUID</td>
</tr>
<tr>
<td>VM-BIOS-UUID</td>
<td>virtual machine</td>
<td>Virtual machine response</td>
</tr>
</tbody>
</table>
Establishing a VMware Telnet Extension Session

If you specify a proxy URI in the virtual serial port configuration, the virtual machine will initiate a telnet connection with the proxy. After the connection is established, the virtual machine initiates a message exchange to establish a VMware telnet extension session. The following progression describes the model for the message exchange and uses symbols to represent the codes. For the code equivalents, see the command descriptions in Chapter 2, “VMware Telnet Extension Commands for Proxy Communication,” on page 17.

1. Start the VMware telnet session.
   - **Virtual Machine** – Sends a message that indicates it will send VMware telnet extension commands to the proxy.
     
     IAC WILL VMware-Telnet-Ext
   
   - **Proxy** – Responds, indicating that it will receive VMware telnet extension commands.
     
     IAC DO VMware-Telnet-Ext

2. Identify known suboptions.
   - **Virtual Machine** – Sends a message that indicates the set of suboption commands that it can send and receive. The suboptions are expressed as a sequence of codes that correspond to suboption commands.
     
     IAC SB VMware-Telnet-Ext KNOWN-SUBOPTIONS-1 suboptions IAC SE
   
   - **Proxy** – Responds, sending a message that indicates the set of suboption commands that it can send and receive.
     
     IAC SB VMware-Telnet-Ext KNOWN-SUBOPTIONS-2 suboptions IAC SE

3. Negotiate proxy support.
   - **Virtual Machine** – Sends a request for the proxy to act as a VMware telnet extension proxy. The proxy request includes direction and URI information from the network backing for the virtual serial port. (For information about using the API to specify this information, see “Creating a Virtual Machine with a Network Serial Port” on page 11.)
     
     IAC SB VMware-Telnet-Ext DO-PROXY direction uri IAC SE
   
   - **Proxy** – Responds, indicating that it will perform proxy operations.
     
     IAC SB VMware-Telnet-Ext WILL-PROXY IAC SE

vMotion and Persistent Serial Port Connections

A virtual serial port proxy provides support for persistent remote serial port connections when a vCenter server moves a virtual machine to a different host. To support the persistent connection during the vMotion operation, the proxy must support two telnet connections for the virtual machine.

A proxy buffers data while it is forwarding content from the remote system to a virtual machine. A proxy must respond to the VMOTION-BEGIN command, which is notification of a pending vMotion operation. ESX Server support for the VMware telnet extension allows the proxy to postpone the vMotion event until it finishes forwarding content. After sending the VMOTION-GOHEAD command in response to VMOTION-BEGIN, the proxy must buffer any additional data it receives from the remote system. When the vMotion event is complete, the proxy continues the content transmission to the new instance of the virtual machine.

---

**Table 1-1. VMware Telnet Extension Commands (Continued)**

<table>
<thead>
<tr>
<th>Command</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET-VM-LOCATION-UUID</td>
<td>proxy</td>
<td>Proxy request for the virtual machine location UUID</td>
</tr>
<tr>
<td>VM-LOCATION-UUID</td>
<td>virtual machine</td>
<td>Virtual machine response</td>
</tr>
</tbody>
</table>
Figure 1-3 shows two ESX hosts and a proxy at the start of a vMotion operation. Host A has a virtual machine that will be moved to Host B.

**Figure 1-3. VMOTION-BEGIN and VMOTION-GOAHEAD**

- **Virtual Machine** – At the beginning of the vMotion operation, the ESX server (Host A) uses the virtual machine’s serial port connection to send a VMOTION-BEGIN request to the proxy. To identify the virtual machine, the host provides an opaque sequence value for the message.

  IAC SB VMWARE-TELNET-EXT VMOTION-BEGIN sequence IAC SE

- **Proxy** – After sending pending data, the proxy replies by sending a VMOTION-GOAHEAD message to indicate that the vMotion operation can continue. The message includes the VMOTION-BEGIN sequence value and an opaque secret value, which the proxy provides.

  IAC SB VMWARE-TELNET-EXT VMOTION-GOAHEAD sequence secret IAC SE

After the proxy replies with a VMOTION-GOAHEAD response, the vMotion operation begins. At this point, there is a single connection between the virtual machine and the proxy.

The vCenter server creates a second instance of the virtual machine on Host B. When the virtual machine boots on the Host B, it configures network backing for the virtual serial port and establishes a second telnet connection with the proxy. Before continuing with the vMotion operation, the new virtual machine and the proxy renegotiate the telnet COM-PORT-OPTION. They do not renegotiate the Com Port configuration. The proxy should be prepared to support the same Com Port configuration that was established for the original telnet connection. See RFC2217 for information about the telnet com port control option. The proxy now maintains one telnet connection for each instance of the virtual machine.

To start the VMware telnet extension session for the new connection, the new virtual machine instance negotiates the VMware telnet extension option (VMWARE-TELNET-EXT).

**Figure 1-4** shows the systems during the vMotion operation.

- **Virtual Machine**: To continue the vMotion operation, Host B sends a VMOTION-PEER message to the proxy to identify the new instance as the same virtual machine that started the vMotion operation. The message includes both the sequence and secret values to identify the virtual machine instance.

  IAC SB VMWARE-TELNET-EXT VMOTION-PEER sequence secret IAC SE

- **Proxy**: The proxy replies with a VMOTION-PEER-OK message to indicate that it accepts the peer connection.

  IAC SB VMWARE-TELNET-EXT VMOTION-PEER-OK sequence IAC SE
Figure 1-4. VMOTION-PEER and VMOTION-PEER-OK

Figure 1-5 shows the completed vMotion operation. After the proxy accepts the new virtual machine instance as a peer, the ESX server on Host B sends a VMOTION-COMPLETE message to the proxy. The message indicates that the proxy should use the new connection for all traffic to the serial port. Now, the proxy can terminate the original telnet connection to Host A.

IAC S8 VMWARE-TELNET-EXT VMOTION-COMPLETE sequence IAC SE

Figure 1-5. VMOTION-COMPLETE
This chapter includes the following sections:
- “Command Syntax” on page 17
- “VMware Telnet Extension Option” on page 18
- “Option Subnegotiation” on page 18
- “Proxy Operation” on page 20
- “vMotion Notification” on page 20
- “Virtual Machine Identification” on page 22

Command Syntax

The VMware telnet extension defines a telnet proxy option for sending notification of vSphere vMotion events to a virtual serial port proxy. A VMware telnet extension command sequence uses the following general format:

```
IAC telnet-command VMWARE-TELNET-EXT [ext-command [parameters] IAC SE]
```

Where
- IAC is the telnet escape character prefix (Interpret as Command).
- `telnet-command` is one of the following telnet commands: WILL, WON’T, DO, DON’T, and SB.
- `VMWARE-TELNET-EXT` is the VMware telnet extension command.
- `ext-command` is a VMware telnet extension command that indicates one of the extension suboption commands. A suboption command sequence begins with the `VMWARE-TELNET-EXT` command and is enclosed within IAC SB ... IAC SE telnet subnegotiation commands.
- `parameters` indicates one or more parameter values.
- IAC SE is the telnet sequence for completing a subnegotiation command sequence.

The following sections contain tables of VMware telnet extension commands. A table entry shows the command, the corresponding integer code, a description, and the command and code sequences. A code sequence for a VMware telnet extension command has the following characteristics.
- The code sequence begins with code 255, the telnet escape character prefix IAC (Interpret as Command).
- VMware telnet extension commands use two- or three-byte code sequences.
- The `VMWARE-TELNET-EXT` command code is 232.
- Subnegotiation command sequences include begin (SB) and end (SE) codes (250 and 240).
For example, the DO–PROXY command requires the following code sequence with embedded arguments for direction (“S” for server) and URI:

```
IAC SB VMWARE–TELNET–EXT DO–PROXY direction uri IAC SE
255 250 232 70 "S" "telnet://example.com" 255 240
```

### VMware Telnet Extension Option

Table 2-1 lists the commands that establish the VMware telnet extension communication between a virtual machine and a proxy. The virtual machine initiates the connection by sending a WILL VMWARE–TELNET–EXT message. If the proxy accepts by responding with a DO VMWARE–TELNET–EXT message, the virtual machine and the proxy can continue negotiation.

**Table 2-1. VMware Telnet Extension Command Negotiation**

<table>
<thead>
<tr>
<th>Command</th>
<th>Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>WILL VMWARE–TELNET–EXT</td>
<td>251 232</td>
</tr>
<tr>
<td>A virtual machine request indicating that it is willing to send VMware telnet extension messages to the proxy.</td>
<td></td>
</tr>
<tr>
<td>IAC WILL VMWARE–TELNET–EXT</td>
<td>255 251 232</td>
</tr>
<tr>
<td>WON’T VMWARE–TELNET–EXT</td>
<td>252 232</td>
</tr>
<tr>
<td>A virtual machine notification that it is unwilling to send VMware telnet extension messages to the proxy. The virtual machine can use this command to turn off the extension option. To resume message exchange, the virtual machine and proxy must negotiate to turn on the option again (WILL VMWARE–TELNET–EXT and DO VMWARE–TELNET–EXT).</td>
<td></td>
</tr>
<tr>
<td>IAC WON’T VMWARE–TELNET–EXT</td>
<td>255 252 232</td>
</tr>
<tr>
<td>DO VMWARE–TELNET–EXT</td>
<td>253 232</td>
</tr>
<tr>
<td>A proxy response that indicates it is willing to receive VMware telnet extension messages from the virtual machine.</td>
<td></td>
</tr>
<tr>
<td>IAC DO VMWARE–TELNET–EXT</td>
<td>255 253 232</td>
</tr>
<tr>
<td>DON’T VMWARE–TELNET–EXT</td>
<td>254 232</td>
</tr>
<tr>
<td>A proxy response that indicates it is unwilling to receive VMware telnet extension messages. The proxy can send this message at any time to stop remote serial port communication. To resume message exchange, the virtual machine and proxy must negotiate to turn on the option again (WILL VMWARE–TELNET–EXT and DO VMWARE–TELNET–EXT).</td>
<td></td>
</tr>
<tr>
<td>IAC DON’T VMWARE–TELNET–EXT</td>
<td>255 254 232</td>
</tr>
</tbody>
</table>

### Option Subnegotiation

The VMWARE–TELNET–EXT telnet option requires various suboptions to support remote serial port access. The virtual machine and proxy use subnegotiation to indicate the command sets that they can support. Subnegotiation identifies all commands for negotiation as well as commands for proxy support, vMotion notification, and virtual machine identification. Table 2-2 lists the commands for option subnegotiation.
### Table 2-2. Option Subnegotiation

<table>
<thead>
<tr>
<th>Command</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNOWN-SUBOPTIONS-1 <code>suboptions</code></td>
<td>0</td>
</tr>
<tr>
<td>KNOWN-SUBOPTIONS-2 <code>suboptions</code></td>
<td>1</td>
</tr>
</tbody>
</table>

A virtual machine notification that indicates the VMWARE-TELNET-EXT suboptions that it can generate and accept. `suboptions` is a sequence of one or more codes, each of which corresponds to a VMware telnet extension command.

If the virtual machine sends this message more than once during a session, the list of suboptions is always the same. The presence of a command in the list does not imply that the virtual machine will use the command during the session.

IAC SB VMWARE-TELNET-EXT KNOWN-SUBOPTIONS-1 `suboptions` IAC SE
255 250 232 0 `suboptions` 255 240

IAC SB VMWARE-TELNET-EXT KNOWN-SUBOPTIONS-2 `suboptions` IAC SE
255 250 232 1 `suboptions` 255 240

### Unknown Command Response

Table 2-3 lists the commands to indicate reception of an unknown command. The proxy must send UNKNOWNSUBOPTION-RCVD-2 any time it receives a command that it does not support.

### Table 2-3. Unknown Suboptions

<table>
<thead>
<tr>
<th>Command</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNKNOWNSUBOPTION-RCVD-1 <code>code</code></td>
<td>2</td>
</tr>
<tr>
<td>UNKNOWNSUBOPTION-RCVD-2 <code>code</code></td>
<td>3</td>
</tr>
</tbody>
</table>

A virtual machine response to indicate that it received an unknown suboption command from the proxy. `code` identifies the unknown suboption.

A proxy response to indicate that it received an unknown suboption command from the virtual machine. `code` identifies the unknown suboption. The proxy must support this command.

IAC SB VMWARE-TELNET-EXT UNKNOWNSUBOPTION-RCVD-1 `code` IAC SE
255 250 232 2 `code` 255 240

IAC SB VMWARE-TELNET-EXT UNKNOWNSUBOPTION-RCVD-2 `code` IAC SE
255 250 232 3 `code` 255 240
Proxy Operation

Table 2-4 lists the extension commands to negotiate proxy support.

<table>
<thead>
<tr>
<th>Command</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>DO–PROXY direction serviceUri</td>
<td>70</td>
</tr>
</tbody>
</table>

A virtual machine request for the proxy to do forwarding operations over the connection. The virtual machine sends a DO–PROXY request before any content is exchanged between the virtual machine and the remote system.

*direction* and *serviceUri* correspond to the VirtualSerialPortURIBackingInfo properties in the virtual serial port configuration. (See the vSphere API Reference for information about the VirtualSerialPortURIBackingInfo data object.)

*direction* is either “C” (client) or “S” (server). If the direction indicates that the virtual machine is acting as a client, *serviceUri* identifies a remote system and the proxy initiates a connection with the remote system. If the direction indicates that the virtual machine is acting as a server, *serviceUri* identifies the network socket on the host on which the virtual machine runs. In this case, the virtual machine will accept a connection initiated by a remote system (using the proxy).

If possible, the proxy should also handle port control options, as specified by RFC 2217.

```
IAC SB VMware-Telnet-Ext DO–PROXY direction uri IAC SE
255 250 232 70 direction uri 255 240
```

WILL–PROXY

A proxy response to indicate that it will do forwarding operations.

```
IAC SB VMware-Telnet-Ext WILL–PROXY IAC SE
255 250 232 71 255 240
```

WON’T–PROXY

A proxy response to indicate that it will not do forwarding operations.

```
IAC SB VMware-Telnet-Ext WON’T–PROXY IAC SE
255 250 232 73 255 240
```

vMotion Notification

Table 2-5 lists the commands that hosts and proxy use during a vMotion operation. In the following descriptions, the *source* host contains the original virtual machine instance, and the *target* host is the host on which the new instance is created. A host uses the virtual machine serial port connection with the proxy to send vMotion messages.
Table 2-5. vMotion Communication

<table>
<thead>
<tr>
<th>Command</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMOTION-BEGIN sequence</td>
<td>40</td>
</tr>
<tr>
<td>A source host request to indicate that a vMotion operation is pending. The operation requires a VMOTION-GOAHEAD response to start. The intent is to flush all pending data transmission on the connection. The host provides an opaque sequence value. All subsequent vMotion messages related to the connection between the virtual machine and this proxy use this sequence value.</td>
<td></td>
</tr>
<tr>
<td>IAC SB VMware-TELNET-EXT VMOTION-BEGIN sequence IAC SE</td>
<td></td>
</tr>
<tr>
<td>255 250 232 40 sequence 255 240</td>
<td></td>
</tr>
<tr>
<td>VMOTION-GOAHEAD sequence secret</td>
<td>41</td>
</tr>
<tr>
<td>A proxy response to indicate that all queued data has been transmitted, and that the vMotion operation can start. The proxy sends this message only in response to a VMOTION-BEGIN request. The proxy returns the sequence value that it received in the VMOTION-BEGIN request. The proxy also creates and returns an opaque secret value. The virtual machine uses the sequence and secret to identify itself when it establishes a proxy connection from the new host. See the description of VMOTION-PEER.</td>
<td></td>
</tr>
<tr>
<td>IAC SB VMware-TELNET-EXT VMOTION-GOAHEAD sequence secret IAC SE</td>
<td></td>
</tr>
<tr>
<td>255 250 232 41 sequence secret 255 240</td>
<td></td>
</tr>
<tr>
<td>VMOTION-NOTNOW sequence</td>
<td>43</td>
</tr>
<tr>
<td>A proxy response to indicate that the vMotion operation should not start at this time. This message includes the sequence value that the proxy obtained in the VMOTION-BEGIN request. The proxy can send this message only in response to a VMOTION-BEGIN request.</td>
<td></td>
</tr>
<tr>
<td>IAC SB VMware-TELNET-EXT VMOTION-NOTNOW sequence IAC SE</td>
<td></td>
</tr>
<tr>
<td>255 250 232 43 sequence 255 240</td>
<td></td>
</tr>
<tr>
<td>VMOTION-PEER sequence secret</td>
<td>44</td>
</tr>
<tr>
<td>A target host request to indicate that this connection belongs to the virtual machine associated with sequence and secret. The sequence and secret identify the duplicate connections between the proxy and virtual machine instances during a vMotion operation. The original host generates the sequence value for the VMOTION-BEGIN message. The proxy generates the secret value for the VMOTION-GOAHEAD message.</td>
<td></td>
</tr>
<tr>
<td>IAC SB VMware-TELNET-EXT VMOTION-PEER sequence secret IAC SE</td>
<td></td>
</tr>
<tr>
<td>255 250 232 44 sequence secret 255 240</td>
<td></td>
</tr>
<tr>
<td>VMOTION-PEER-OK sequence</td>
<td>45</td>
</tr>
<tr>
<td>A proxy response to indicate that the secret in the VMOTION-PEER message was accepted. The proxy sends this message only in response to a VMOTION-PEER request.</td>
<td></td>
</tr>
<tr>
<td>IAC SB VMware-TELNET-EXT VMOTION-PEER-OK sequence IAC SE</td>
<td></td>
</tr>
<tr>
<td>255 250 232 45 sequence 255 240</td>
<td></td>
</tr>
<tr>
<td>VMOTION-COMPLETE sequence</td>
<td>46</td>
</tr>
<tr>
<td>A target host notification that the vMotion operation is complete. The proxy must send all subsequent communication for the virtual machine on the connection associated with this VMOTION-COMPLETE message. The host sends this message only after sending a VMOTION-PEER message.</td>
<td></td>
</tr>
<tr>
<td>IAC SB VMware-TELNET-EXT VMOTION-COMPLETE sequence IAC SE</td>
<td></td>
</tr>
<tr>
<td>255 250 232 46 sequence 255 240</td>
<td></td>
</tr>
<tr>
<td>VMOTION-ABORT</td>
<td>48</td>
</tr>
<tr>
<td>A source host notification that the vMotion operation failed. The proxy must send all communication for the virtual machine on the connection associated with this VMOTION-ABORT message. The host can send VMOTION-ABORT only if it has sent VMOTION-BEGIN previously on this connection.</td>
<td></td>
</tr>
<tr>
<td>IAC SB VMware-TELNET-EXT VMOTION-ABORT IAC SE</td>
<td></td>
</tr>
<tr>
<td>255 250 232 48 255 240</td>
<td></td>
</tr>
</tbody>
</table>
Virtual Machine Identification

Table 2-6 lists the suboption commands for virtual machine identification.

### Table 2-6. VMware Telnet Extension Commands for Virtual Machine Identification

<table>
<thead>
<tr>
<th>Command</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM–VC–UUID  \textit{vm–uuid}</td>
<td>80</td>
</tr>
<tr>
<td>GET–VM–VC–UUID</td>
<td>81</td>
</tr>
<tr>
<td>A proxy request for the virtual machine's universally unique identifier.</td>
<td></td>
</tr>
<tr>
<td>VM–NAME  \textit{vm–name}</td>
<td>82</td>
</tr>
<tr>
<td>IAC SB VMware–TELNET–EXT VM–NAME \textit{vm–name} IAC SE 255 250 232 82 \textit{vm–name} 255 240</td>
<td></td>
</tr>
<tr>
<td>GET–VM–NAME</td>
<td>83</td>
</tr>
<tr>
<td>A proxy request for the virtual machine's name.</td>
<td></td>
</tr>
<tr>
<td>VM–BIOS–UUID  \textit{vm–uuid}</td>
<td>84</td>
</tr>
<tr>
<td>GET–VM–BIOS–UUID</td>
<td>85</td>
</tr>
<tr>
<td>A proxy request for the virtual machine's universally unique identifier stored in its BIOS.</td>
<td></td>
</tr>
<tr>
<td>VM–LOCATION–UUID  \textit{vm–uuid}</td>
<td>86</td>
</tr>
<tr>
<td>GET–VM–LOCATION–UUID</td>
<td>87</td>
</tr>
<tr>
<td>A proxy request for the virtual machine's location universally unique identifier.</td>
<td></td>
</tr>
</tbody>
</table>
# Index

## A
- accepting a connection 11
- access server 7
- access, virtual machine 8
- appliance, virtual 8
- authentication 8

## B
- backing info object 11
- backing options 9, 11

## C
- command syntax 17
- commands
  - DO-PROXY 20
  - GET-VM-BIOS-UUID 22
  - GET-VM-LOCATION-UUID 22
  - GET-VM-NAME 22
  - GET-VM-VC-UUID 22
  - KNOWN-SUBOPTIONS-1 19
  - KNOWN-SUBOPTIONS-2 19
  - UNKNOWN-SUBOPTIONS-RCVD-1 19
  - UNKNOWN-SUBOPTIONS-RCVD-2 19
  - VM-BIOS-UUID 22
  - VM-LOCATION-UUID 22
  - VM-NAME 22
  - VMOTION-ABORT 21
  - VMOTION-BEGIN 21
  - VMOTION-COMPLETE 21
  - VMOTION-GOHEAD 21
  - VMOTION-NOTNOW 21
  - VMOTION-PEER 21
  - VMOTION-PEER-OK 21
  - VM-VC-UUID 22
  - VMware telnet code 17
  - VMware telnet extension 12
  - VMware-TELNET-EXT 18
  - WILL-PROXY 20
  - WON'T-PROXY 20
- COM-PORT-OPTION (telnet) 14
- concentrator 7
- configuring a network serial port 11
- connection
  - behavior 11
  - direction 11

## D
- data object, configuration 11
- direction 11, 20
  - connection 9
  - server 11
- DO-PROXY 20

## E
- establishing a VMware telnet extension session 13

## F
- forwarding traffic 8

## G
- GET-VM-BIOS-UUID 22
- GET-VM-LOCATION-UUID 22
- GET-VM-NAME 22
- GET-VM-VC-UUID 22

## H
- host IP address 7

## I
- IAC (interpret as command) 17
- IP address
  - host 7
  - virtual machine 8

## K
- KNOWN-SUBOPTIONS-1 19
- KNOWN-SUBOPTIONS-2 19

## M
- method, configuration 11

## N
- negotiation, VMware telnet session 12

## O
- option
  - backing (configuration) 11
  - subnegotiation (telnet) 18

## P
- proxy 7, 9
  - connection behavior 11
  - third party 7
Using a Proxy with vSphere Virtual Serial Ports

URI 11
  virtual appliance 8

Q
  QueryConfigOption method 11

R
  ReconfigVM_Task method 11
  RFC 2217 8, 20
  3986 9
  role of the virtual serial port proxy 8

S
  secret (VMOTION-GOAHEAD) 21
  security 8
  sequence (VMOTION-BEGIN) 21
  server direction 11
  serviceURI 9
    backing info 11
    DO-PROXY 20
  session negotiation 12
  syntax,command 17

T
  telnet
    COM-PORT-OPTION 14
    connections for vMotion 13
    encrypted connection 9
    extension 7, 12
    option subnegotiation 18
    session 13
    session negotiation 12
  third party proxy 7

U
  UNKNOWN-SUBOPTIONS-RCVD-1 19
  UNKNOWN-SUBOPTIONS-RCVD-2 19
  URI
    backing option 9
    proxy 9, 11
    service 9, 11

V
  virtual appliance 8
  virtual machine
    identification 22
    IP address 8
    serial port client 9
    serial port server 8, 9
  virtual serial port concentrator 7
  VirtualDeviceConfigSpec data object 11
  VirtualDeviceConnectInfo data object 11
  VirtualDeviceURIBackingInfo data object 11
  VirtualDeviceURIBackingOption data object 11
  VirtualMachineConfigOption data object 11
  VirtualMachineConfigSpec data object 11
  VirtualSerialPort data object 11
  VM-BIOS-UUID 22
  VM-LOCATION-UUID 22
  VM-NAME 22
  vMotion 8
  persistent serial port connections 13
  VMOTION-ABORT 21
  VMOTION-BEGIN 21
  VMOTION-COMPLETE 21
  VMOTION-GOAHEAD 21
  VMOTION-NOTNOW 21
  VMOTION-PEER 21
  VM-VC-UUID 22
  VMware telnet
    extension command code 17
    extension commands 12
    session negotiation 12
  VMWARE-TELNET-EXT 18

W
  WILL-PROXY 20
  WON'T-PROXY 20