

Serial Device Configuration  
on SRE-V with Linux  
OS Virtual Machine



# Serial Device Configuration on SRE-V

## Table of Contents

Introduction .....	3
Router Hardware and Interface Configuration .....	4
Serial GPS .....	5
Serial Barcode Reader.....	9
Point-of-Sale Printer .....	12
Uninterruptable Power Supply .....	15
Appendix A: Installing and Configuring Cyclades .....	19



# Serial Device Configuration on SRE-V

## Introduction

The Service Ready Engine Virtualization (SRE-V) platform leverages VMware vSphere Hypervisor to provide full virtualization so that multiple Virtual Machines (VM) and their operating systems (OS) may be installed and run on a Service Ready Engine (SRE) module that plugs into the second generation Integrated Service Router (ISR). Please see the SRE-V Installation and Configuration Guide for a list of supported SRE and ISR devices.

This whitepaper addresses serial device control from the VM containing a Linux operating system. Specifically four different types of serial devices are covered:

1. Global Positioning System (GPS)
2. Barcode Reader
3. Point of Sale (POS) Printer
4. Uninterruptable Power Supply (UPS)

A separate topic is provided for each specific device with detailed instructions given for configuration and implementation within a specified Linux environment.

## Assumptions

1. SRE-V release 2.0.1 or higher is installed on a SRE module that is plugged into an ISR G2 router.
2. The ISR IOS image is 15.1(4)M or higher
3. A running Virtual Machine (VM) is configured on the SRE-V platform with the 64 bit Linux platform CentOS release 5.6. The VM can ping the router, and is accessible via the network.
4. To connect to the first three devices discussed: GPS, Barcode Reader and POS Printer, the ISR G2 router has an Hi-Speed WAN Interface Card (HWIC) plugged in that supports serial device RS232 connection (example HWIC 1T or HWIC 2T).
5. For the last device discussed: UPS, a Serial to USB cable is required.
6. The user has a strong working knowledge of VMware vSphere Hypervisor.
7. For configuration of the first three devices the user is familiar with the configuration and use of minicom.

## Disclaimer

The instructions provided explain how to connect to the serial devices and to read from/write to them on a basic level. The interpretation of the read data and composition of more complex commands are left to the user and not within the scope of this paper. Serial devices of different brands and models not tested in the whitepaper may not function properly and the configuration for the communication to such devices is not supported.

Cisco does not endorse or provide end customer support for any of the serial devices, Cyclades, CENTOS or any other products not developed and sold by Cisco. Customers should contact the associated vendors for support of third party products.

# Serial Device Configuration on SRE-V

## Router Hardware and Interface Configuration

The hardware inspection and interface configuration steps below apply only to the GPS, Barcode Reader and POS Printer serial devices covered in this white paper. Make sure to follow these steps first before beginning the set up for these serial devices.

- Step 1** Check that the ISR contains an HWIC card that supports serial device connection. On the router prompt type: **show diag | inc FRU** . In the displayed data an HWIC card supporting serial device connection should be reported.

```
3925Router#show diag | inc FRU
Product (FRU) Number : CISCO3925-CHASSIS
Product (FRU) Number : C3900-SPE100/K9
Product (FRU) Number : PWR-3900-AC
Product (FRU) Number : HWIC-1T
Product (FRU) Number : SM-SRE-900-K9
```

**Figure 1: Checking Hardware**

- Step 2** Configure the serial interface in the router. An example configuration is shown below.

```
interface Serial0/0/0
physical-layer async
no ip address
encapsulation slip
end
```

**Figure 2: Configuring the Interface**

- Step 3** The router command **show ip interface brief** will show both the status and protocol as ‘down’ for the serial interface. This status is ok.

- Step 4** The router is now ready for serial device use.



# Serial Device Configuration on SRE-V

## Serial GPS

In this configuration we will be reading the global positioning data provided by a Garmin GPS 18 PC serial device. In our lab we connected the GPS device to a 25 Pin-to-9 Pin Cisco adapter Model 29-4043-01 V-1. We then used an RS 232/V.24 DCE cable to connect from the adapter to the HWIC-1T card plugged into the ISR.

We utilized the open source third party software Cyclades Serial Client to make the GPS device accessible within the VM via a specified /dev file. In our tests we found that when Cyclades initially receives data from the GPS device it will log a buffer overrun error in the file /var/log/messages; however, this overrun lasts just a few seconds and the data that follows is received as expected.

The steps below provide the configuration and testing procedures necessary to assure the proper connection to the GPS device in both the ISR and the VM. Before beginning the steps below first make sure that you've completed the configuration steps in the section: "Router Hardware and Interface Configuration".

**Step 1** Connect the serial GPS device to the HWIC card. Power on the GPS device.

**Step 2** On the router type **show line** and check for the interface. Observe the line number and the transfer and receive speeds. An example is provided in Figure 1. Note that in this example the line number is 3, and the transfer and receive speeds are 4800/4800.

```
3925Router#show line
```

Tty	Line	Typ	Tx/Rx	A	Modem	Roty	AccO	AccI	Uses	Noise	Overruns	Int
0	0	CTY		-	-	-	-	-	0	0	0/0	-
1	1	AUX	9600/9600	-	-	-	-	-	0	0	0/0	-
2	2	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
0/0/0	3	TTY	4800/4800	-	-	-	-	-	0	48	31/0	Se0/0/0

**Figure 1: Gathering Device Information**

**Step 3** From the VM console in vSphere verify that the GPS device is sending information. In the telnet command you must specify the IP or hostname of your router to which the serial GPS device is connected along with the serial port line number discussed in Step 2. Add 6000 to the line number for serial device communication. An example is **telnet 192.168.1.70 6003** . You should receive continuous output similar to the format shown in Figure 2 below. Exit the telnet session.

# Serial Device Configuration on SRE-V



```
[cisco@localhost ~]$ telnet 192.168.1.70 6003
Trying 192.168.1.70...
Connected to 192.168.1.70 (192.168.1.70).
Escape character is '^'.
,,M,,M,,*4E
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGSV,3,1,12,09,52,235,00,10,01,160,00,11,00,039,00,12,50,318,00*73
$GPGSV,3,2,12,13,00,084,00,14,01,309,00,15,00,203,00,16,00,287,00*74
$GPGSV,3,3,12,17,18,063,00,18,00,244,00,19,00,106,00,20,00,026,00*72
$PGRME,,M,,M,,M*00
$GPGLL,3725.0112,N,12155.1412,W,102351,V*2E
$GPVTG,,T,,M,,N,,K*4E
$PGRMV,,,*72
$PGRMF,,230408,102351,,3725.0112,N,12155.1412,W,A,0,,,*36
$PGRMB,,,,,K,,,*2D
$PGRMM,WGS 84*06
$GPRMC,102352,V,3725.0112,N,12155.1412,W,,,230408,014.9,E*70
```

**Figure 2: Checking GPS Device Output**

**Step 4** Complete the steps specified in Appendix A: Installing and Configuring Cyclades.

**Step 5** Edit the file `/etc/cyclades-devices` to make the changes below.

- a. Comment the modem device line below.
  - i. `#/dev/modem:path:localhost:1:socket:`
- b. Below that line add the gps device connection information. You must specify the device name, the IP or hostname of your router to which the serial GPS device is connected along with the line number in the format: **<Device Location and Name>:path:<Router IP or Host Name>:<line number>:rtelnet** . An example is provided below.
  - i. `/dev/gps:path:192.168.1.70:3:rtelnet`
- c. Save the file.

```
# Ex3 : Device on a PathRAS Terminal Server pras Port 2, accessed through
# /dev/tty02 device, using socket protocol (socket server)
#
#/dev/tty02:path:pras:2:socket:
# for Linux
#/dev/modem:path:localhost:1:socket:
/dev/gps:path:192.168.1.70:3:rtelnet
```

**Figure 3: Edit /etc/cyclades-devices**

**Step 6** With the changes made in Step 5, when activated Cyclades will create a virtual GPS device `/dev/gps` from which the GPS relayed data can be read. As user **root** start the Cyclades application then verify that it is running and that the virtual devices have been created. An example is shown below in Figure 4. If data returned from the 'status' command shows the pid as 'inactive' check the `/var/log/messages` file for the cause.

# Serial Device Configuration on SRE-V



```
[root@localhost sbin]# ./cyclades-serial-client start
Starting /dev/gps <<==>> 192.168.1.70:3 interface

[root@localhost sbin]# ./cyclades-serial-client status
/dev/gps (rtelnet at 192.168.1.70:3) active, pid 6372

[root@localhost sbin]# ls -l /dev/gps
lrwxrwxrwx 1 root root 10 Dec 14 15:10 /dev/gps -> /dev/pts/3

[root@localhost sbin]# ls -l /dev/pts/3
crw--w---- 1 root tty 136, 3 Dec 14 15:10 /dev/pts/3
```

**Figure 4: Starting Cyclades**

**Step 7** The application ‘minicom’ can be configured to view the data being received by the `/dev/gps` device. In our tests we had success with the configuration parameters shown in Figure 5. In Figure 6, we’ve provided a snippet of the data viewed via minicom.

```
Serial Device      : /dev/gps
Bps/Par/Bits      : 4800 8N1
Hardware Flow Control : No
Software Flow Control : No
```

**Figure 5: Minicom Configuration**

```
$GPGGA,122033,3725.0112,N,12155.1412,W,0,00,,M,,M,,*4B
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGSV,3,1,12,10,43,122,00,11,00,070,00,12,73,172,00,13,00,043,00*7E
$GPGSV,3,2,12,14,00,266,00,15,00,162,00,16,00,323,00,17,00,098,00*7F
$GPGSV,3,3,12,18,00,205,00,19,00,229,00,20,00,002,00,21,07,232,00*72
$PGRME,,M,,M,,M*00
$GPGLL,3725.0112,N,12155.1412,W,122033,V*2B
$GPVTG,,T,,M,,N,,K*4E
$PGRMV,,,*72
$PGRMF,,230408,122033,,3725.0112,N,12155.1412,W,A,0,,,*33
$PGRMB,,,,,K,,*2D
$PGRMM,WGS 84*06
$GPRMC,122034,V,3725.0112,N,12155.1412,W,,230408,014.9,E*71
```

**Figure 6: Snippet of /dev/gps Device Data via Minicom**

**Step 8** Cyclades can also be shutdown and its status rechecked to verify it is down. See Figure 7 below.

```
[root@localhost sbin]# ./cyclades-serial-client stop
Stopping /dev/gps <<==>> 192.168.1.70:3 interface

[root@localhost sbin]# ./cyclades-serial-client status
/dev/gps (rtelnet at 192.168.1.70:3) inactive
```

**Figure 7: Cyclades Shutdown and Status**

# Serial Device Configuration on SRE-V



**Step 9** The configuration is completed.



# Serial Device Configuration on SRE-V

## Serial Barcode Reader

In this configuration we will be reading the barcode information processed by a Motorola Symbol LS2208 Barcode Reader device. This device contains an RS232 kit. The provided cable connects to the bar code device and has a 9 Pin connector. In our lab we connected the Barcode Reader cable to a 25 Pin-to-9 Pin Cisco adapter Model 29-4043-01 V-1. We then used an RS 232/V.24 DCE cable to connect from the adapter to the HWIC-1T card plugged into the ISR.

We utilized the open source third party software Cyclades Serial Client to make the GPS device accessible within the VM via a specified /dev file.

The steps below provide the configuration and testing procedures necessary to assure the proper connection to the GPS device in both the ISR and the VM. Before beginning the steps below first make sure that you've completed the configuration steps in the section: "Router Hardware and Interface Configuration".

- Step 1** Connect the serial Barcode Reader device to the HWIC card. Power on the Barcode Reader device. The HWIC card connection light should now be green.
- Step 2** On the router type **show line** and check for the interface. Observe the line number and the transfer and receive speeds. An example is provided in Figure 1. Note that in this example the line number is 3, and the transfer and receive speeds are 9600/9600.

```
3925Router#show line
```

Tty	Line	Typ	Tx/Rx	A	Modem	Roty	AccO	AccI	Uses	Noise	Overruns	Int
0	0	CTY		-	-	-	-	-	0	0	0/0	-
1	1	AUX	9600/9600	-	-	-	-	-	0	0	0/0	-
2	2	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
0/0/0	3	TTY	9600/9600	-	-	-	-	-	0	4	0/0	Se0/0/0

**Figure 1: Gathering Device Information**

- Step 3** From the VM console in vSphere verify that the Barcode Reader device is sending information. In the telnet command you must specify the IP or hostname of your router to which the serial Barcode Reader device is connected along with the serial port line number discussed in Step 2. Add 6000 to the line number for serial device communication. An example is '**telnet 192.168.1.70 6003**'.
- Step 4** Press the trigger of the Barcode Reader. You should see a red horizontal line projecting from the device. Next, attempt to scan a barcode. If there is no horizontal line or the device does not scan the barcode refer to Motorola's Quick Start Guide for the Symbol LS2208 device. There is also helpful information at the following web pages: <http://www.motorola.com/lS2208> and [http://support.symbol.com/support/search.do?cmd=displayKC&docType=kc&externalId=ADC-00424&sliceId=SAL\\_Public](http://support.symbol.com/support/search.do?cmd=displayKC&docType=kc&externalId=ADC-00424&sliceId=SAL_Public).

# Serial Device Configuration on SRE-V



Assuming the device is working as desired, you should see a scanned barcode appear onscreen. An example is shown in Figure 2 below. Exit the telnet session.

```
[cisco@localhost ~]# telnet 192.168.1.70 6003
Trying 192.168.1.70...
Connected to 192.168.1.70 (192.168.1.70).
Escape character is '^]'.
13SB02AGB1324 ← Scanned Barcode
```

**Figure 2: Barcode Reader Device Output**

**Step 5** Complete the steps specified in Appendix A: Installing and Configuring Cyclades.

**Step 6** Edit the file `/etc/cyclades-devices` to make the changes below.

- a. Comment the modem device line below.
  - i. **`#/dev/modem:path:localhost:1:socket:`**
- b. Below that line add the Barcode Reader device connection information. You must specify the device name, the IP or hostname of your router to which the serial Barcode Reader device is connected along with the line number in the format: **<Device Location and Name>;path:<Router IP or Host Name>:<line number>:rtelnet** . An example is provided below.
  - i. **`/dev/barcode:path:192.168.1.70:3:rtelnet`**
- c. Save the file.

```
# Ex3 : Device on a PathRAS Terminal Server pras Port 2, accessed through
# /dev/tty02 device, using socket protocol (socket server)
#
#/dev/tty02:path:pras:2:socket:
# for Linux
#/dev/modem:path:localhost:1:socket:
/dev/barcode:path:192.168.1.70:3:rtelnet
```

**Figure 3: Edit `/etc/cyclades-devices`**

**Step 7** With the changes made in Step 6, when activated Cyclades will create a virtual Barcode Reader device **`/dev/barcode`** from which the Barcode Reader relayed data can be read. As user **root** start the Cyclades application then verify that it is running and that the virtual devices have been created. An example is shown below in Figure 4. If data returned from the 'status' command shows the pid as 'inactive' check the **`/var/log/messages`** file for the cause.

# Serial Device Configuration on SRE-V



```
[root@localhost ~]# /usr/sbin/cyclades-serial-client start
Starting /dev/barcode <<===>> 192.168.1.70:3 interface

[root@localhost ~]# /usr/sbin/cyclades-serial-client status
/dev/barcode (rtnet at 192.168.1.70:3) active, pid 4104

[root@localhost ~]# ls -l /dev/barcode
lrwxrwxrwx 1 root root 10 Dec 21 17:09 /dev/barcode -> /dev/pts/2

[root@localhost ~]# ls -l /dev/pts/2
crw--w---- 1 root tty 136, 2 Dec 21 17:09 /dev/pts/2
```

**Figure 4: Starting Cyclades**

**Step 8** The application ‘minicom’ can be configured to view the data being received by the ‘/dev/barcode’ device. In our tests we had success with the configuration parameters shown in Figure 5. In Figure 6, we’ve provided an example of the data viewed via minicom.

```
Serial Device      : /dev/barcode
Bps/Par/Bits      : 9600 8N1
Hardware Flow Control : No
Software Flow Control : No
```

**Figure 5: Minicom Configuration**

```
Welcome to minicom 2.1

OPTIONS: History Buffer, F-key Macros, Search History Buffer, I18n
Compiled on Jan 7 2007, 01:16:05.

Press CTRL-A Z for help on special keys

FOC14232JL8 ← Scanned Barcode
```

**Figure 6: Example of /dev/barcode Device Data via Minicom**

**Step 9** Cyclades can also be shutdown and its status rechecked to verify it is down. See Figure 7 below.

```
[root@localhost ~]# /usr/sbin/cyclades-serial-client stop
Stopping /dev/barcode <<===>> 192.168.1.70:3 interface

[root@localhost ~]# /usr/sbin/cyclades-serial-client status
/dev/barcode (rtnet at 192.168.1.70:3) inactive
```

**Figure 7: Cyclades Shutdown and Status**

**Step 10** The configuration is completed.

# Serial Device Configuration on SRE-V

## Point-of-Sale Printer

In this configuration we will be sending print data to the EPSON Point of Sale (POS) Serial Receipt Printer TM-U220. The vendor provided DB9 serial cable connects to the POS Printer device and has a 9 Pin connector. In our lab we connected the DB9 serial cable to a 25 Pin-to-9 Pin Cisco adapter Model 29-4043-01 V-1. We then used an RS 232/V.24 DCE cable to connect from the adapter to the HWIC-1T card plugged into the ISR.

We utilized the open source third party software Cyclades Serial Client to make the POS Printer device accessible within the VM via a specified /dev file.

The steps below provide the configuration and testing procedures necessary to assure the proper connection to the POS device in both the ISR and the VM. Before beginning the steps below first make sure that you've completed the configuration steps in the section: "Router Hardware and Interface Configuration".

- Step 1** Connect the serial POS Printer device to the HWIC card. Power on the POS Printer device. The HWIC card connection light should now be green.
- Step 2** On the router type: **show line** and check for the interface. Observe the line number and the transfer and receive speeds. An example is provided in Figure 1. Note that in this example the line number is 3, and the transfer and receive speeds are 9600/9600.

```
3925Router#show line
```

Tty	Line	Typ	Tx/Rx	A	Modem	Roty	AccO	AccI	Uses	Noise	Overruns	Int
0	0	CTY		-	-	-	-	-	0	0	0/0	-
1	1	AUX	9600/9600	-	-	-	-	-	0	0	0/0	-
2	2	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
0/0/0	3	TTY	9600/9600	-	-	-	-	-	0	4	0/0	Se0/0/0

**Figure 1: Gathering Device Information**

- Step 3** From the VM console in vSphere verify that you can send data to the POS Printer device. In the telnet command you must specify the IP or hostname of your router to which the serial POS Printer device is connected along with the serial port line number discussed in Step 2. Add 6000 to the line number for serial device communication. An example is **telnet 192.168.1.70 6003** . See Figure 2 below.
- Step 4** Type **test** at the telnet screen. The text will not appear onscreen, but you should hear the POS Printer print the text. Access the POS printer and press the 'FEED' button until you see the text. If the printer does not print the text refer to the EPSON TM-U220 User's Manual. Exit the telnet session.

# Serial Device Configuration on SRE-V



```
[cisco@localhost ~]# telnet 192.168.1.70 6003
Trying 192.168.1.70...
Connected to 192.168.1.70 (192.168.1.70).
Escape character is '^'.
```

**Figure 2: Telnet to the Port**

**Step 5** Complete the steps specified in Appendix A: Installing and Configuring Cyclades.

**Step 6** Edit the file `/etc/cyclades-devices` to make the changes below.

- a. Comment the modem device line below.
  - i. **`#/dev/modem:path:localhost:1:socket:`**
- b. Below that line add the POS Printer device connection information. You must specify the device name, the IP or hostname of your router to which the serial POS Printer device is connected along with the line number in the format: **<Device Location and Name>:path:<Router IP or Host Name>:<line number>:rtelnet** . An example is provided below.
  - i. **`/dev/pos:path:192.168.1.70:3:rtelnet`**
- c. Save the file.

```
# Ex3 : Device on a PathRAS Terminal Server pras Port 2, accessed through
# /dev/tty02 device, using socket protocol (socket server)
#
#/dev/tty02:path:pras:2:socket:
# for Linux
#/dev/modem:path:localhost:1:socket:
/dev/pos:path:192.168.1.70:3:rtelnet
```

**Figure 3: Edit /etc/cyclades-devices**

**Step 7** With the changes made in Step 6, when activated Cyclades will create a virtual POS Printer device `/dev/pos` to which data can be sent. As user **root** start the Cyclades application then verify that it is running and that the virtual devices have been created. An example is shown below in Figure 4. If data returned from the ‘status’ command shows the pid as ‘inactive’ check the `/var/log/messages` file for the cause.

```
[root@localhost cisco]# /usr/sbin/cyclades-serial-client start
Starting /dev/pos <<===>> 192.168.1.70:3 interface

[root@localhost cisco]# /usr/sbin/cyclades-serial-client status
/dev/pos (rtelnet at 192.168.1.70:3) active, pid 911

[root@localhost cisco]# ls -l /dev/pos
lrwxrwxrwx 1 root root 10 Jan  7 16:07 /dev/pos -> /dev/pts/1

[root@localhost cisco]# ls -l /dev/pts/1
crw--w---- 1 root tty 136, 1 Jan  7 16:07 /dev/pts/1
```

**Figure 4: Starting Cyclades**

# Serial Device Configuration on SRE-V



**Step 8** Now again test that data can be sent to the device by entering the command: **echo test>/dev/pos** . The POS Printer should print the word “test”. Access the printer and press the “FEED” button until you see the text.

```
[root@localhost cisco]# echo test>/dev/pos
```

**Figure 5: Testing Device Output**

**Step 9** Cyclades can also be shutdown and its status rechecked to verify it is down. See Figure 6 below.

```
[root@localhost cisco]# /usr/sbin/cyclades-serial-client stop
Stopping /dev/pos <<==>> 192.168.1.70:3 interface

[root@localhost cisco]# /usr/sbin/cyclades-serial-client status
/dev/pos (rtelnet at 192.168.1.70:3) inactive
```

**Figure 6: Cyclades Shutdown and Status**

**Step 10** The configuration is completed.

# Serial Device Configuration on SRE-V

## Uninterruptable Power Supply

In this configuration we will be connecting to the CyberPower 1000 AVR Uninterruptable Power Supply (UPS) device to determine its status. This connection requires a USB to Serial DB9 cable that must be purchased separately. In our lab we connected the DB9 end of the cable to the UPS device serial port and plugged in the USB side to the SRE module USB port. In order to test the UPS device communication during a power outage we plugged the UPS into a power strip and switched the strip off.

CyberPower provides a client-server software for Linux named PowerPanel for Linux. This software can be downloaded at <http://www.cyberpowersystems.com/products/management-software.html> . In our lab we installed into our CentOS VM the software 'PowerPanel for 64 bit Linux rpm 1.2 (Linux 64 bit)'

The steps below provide the configuration and testing procedures necessary to assure the proper connection to the UPS device in the VM.

- Step 1** Connect the serial UPS device to the USB port on the SRE module via a USB to Serial DB9 cable. Power on the UPS device.
- Step 2** In order for the VM to utilize the USB connection, a USB controller and device must be added to the VM. Log into vSphere, right click on the VM and in the context menu select **Edit Settings ...** . In the **Virtual Machine Properties** window click on the **Hardware** tab. Next click on the **Add...** button. In the **Add Hardware** window click on the **USB Controller** icon and click **Next**. Select the **EHCI + UHCI** controller type shown in Figure 2, and click **Next** .

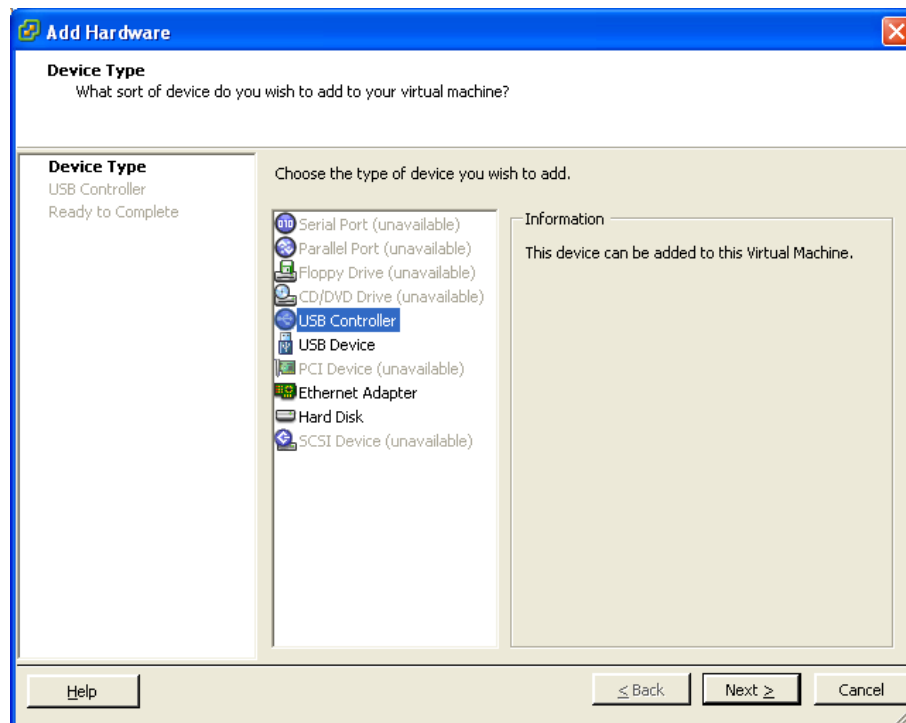
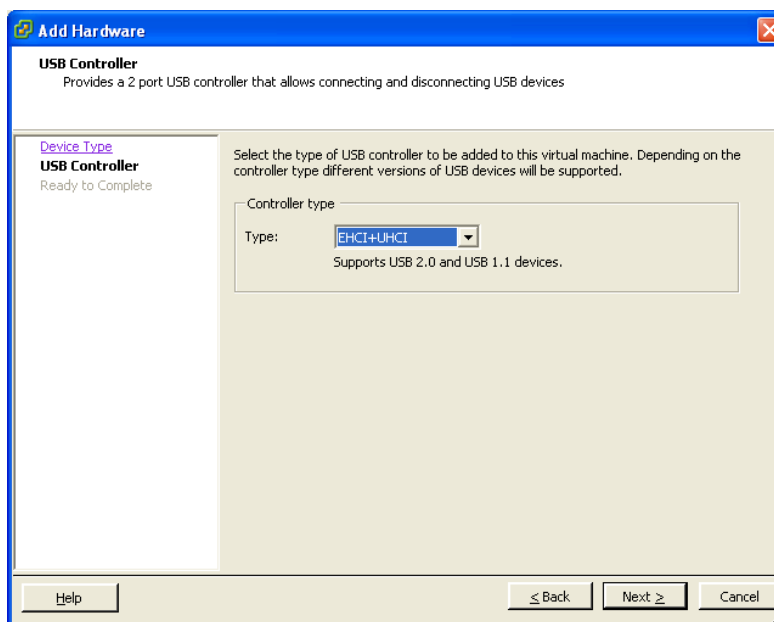


Figure 1: Adding USB Support

## Serial Device Configuration on SRE-V

Complete the controller addition by clicking on the **Finish** button



**Figure 2: USB Controller Type**

- Step 3** The USB device now must be added to the VM. In the **Virtual Machine Properties** window click on the **Hardware** tab. Next click on the **Add...** button. In the **Add Hardware** window click on the **USB Device** icon and click **Next** . Complete adding the device by clicking **Next** on the screen that follows, and finally click **Finish** . With this change, ESXi will now create a device named `/dev/ttyUSB0` . This device is the communication link between the UPS and PowerPanel Server
- Step 4** Open the VM console and then download and install the CyberPower PowerPanel for 64 bit Linux rpm software. This RedHat Package Module (RPM), can be installed from the console via the command: `rpm -ivh powerpanel_1.2_x86_64.rpm` .
- Step 5** Among the files installed by PowerPanel there are a few that should be mentioned. Please review the PowerPanel documentation for more information.
- a. `/etc/pwrstatd.conf`
    - i. The PowerPanel server configuration file.
  - b. `/etc/init.d/pwrstatd`
    - i. The startup script for the PowerPanel server.
  - c. `/usr/sbin/pwrstatd`
    - i. The PowerPanel server.
  - d. `/usr/sbin/pwrstat`
    - i. The PowerPanel client.
  - e. `/var/logs/pwrstatd.log`
    - i. The PowerPanel server log.



## Serial Device Configuration on SRE-V

- Step 6** As user **root** open the PowerPanel server configuration file in **/etc/pwrstatd.conf**. Uncomment the line: **#allowed-device-nodes = ttyS;ttyUSB**, by removing the **#** so that the line will now be active. Save and close the file. This change allows the PowerPanel server to communicate with the UPS device via the **/dev/ttyUSB0** device described earlier.

```
# For example: restrict to use ttyS and ttyUSB two groups of device node at
# /dev,/dev/usb, and /dev/usb/hid paths(includes ttyS0 to ttySN and ttyUSB0 to
# ttyUSBN, N is number).
allowed-device-nodes = ttyS;ttyUSB
```

**Figure 3: Editing the PowerPanel Server Configuration File**

- Step 7** Restart the PowerPanel Server by entering the command: **/etc/init.d/pwrstatd restart** shown in Figure 4. Confirm that the server is running via entering: **/etc/init.d/pwrstatd status** .

```
[root@localhost etc]# /etc/init.d/pwrstatd restart
Reloading pwrstatd 1.2
Shutting down pwrstatd [ OK ]
Starting pwrstatd
[root@localhost etc]# /etc/init.d/pwrstatd status
Status pwrstatd 1.2
pwrstatd (pid 4789) is running...
```

**Figure 4: Restart the PowerPanel Server**

- Step 8** It may take a few minutes for the PowerPanel server to establish communication with the UPS device. The PowerPanel server log file located in **/var/log/pwrstatd.log** provides this server connection information as shown in Figure 5.

```
2012/02/27 15:27:11 PM Daemon stops its service.
2012/02/27 15:27:13 PM Daemon startups.
2012/02/27 15:29:44 PM Communication is established.
```

**Figure 5: Reviewing the PowerPanel Logs**

- Step 9** Run the PowerPanel client to get the UPS status by entering the command: **/usr/sbin/pwrstat -status** . The client returns the status and then terminates.

```
[root@localhost etc]# /usr/sbin/pwrstat -status
The UPS information shows as following:
Current UPS status:
State..... Normal
Power Supply by..... Utility Power
Last Power Event..... Blackout at 2012/02/25 15:30:27
```

**Figure 6: Access the UPS Status**

# Serial Device Configuration on SRE-V

**Step 10** Test that a changed status is returned by shutting down the power strip on which the UPS is plugged in, and again running the client via the command: **usr/sbin/pwrstat -status**

```
[root@localhost etc]# /usr/sbin/pwrstat -status
The UPS information shows as following:
Current UPS status:
State..... Power Failure
Power Supply by..... Battery Power
Last Power Event..... Blackout at 2012/02/28 14:42:08
```

**Figure 7: Check UPS Status Change**

**Step 11** When the power strip on which the UPS resides is powered back on, it may take a few minutes for the server to reestablish communication with the UPS device. As a result the client will temporarily return a Lost Communication status as shown in Figure 8.

```
[root@localhost etc]# /usr/sbin/pwrstat -status
The UPS information shows as following:
Current UPS status:
State..... Lost Communication
Last Power Event..... Blackout at 2012/02/28 14:42:08
```

**Figure 8: Lost Communication**

**Step 12** The PowerPanel server log file located in /var/log/pwrstatd.log shows the server status and communication changes.

```
2012/02/28 13:52:08 PM Daemon startups.
2012/02/28 13:54:39 PM Communication is established.
2012/02/28 14:42:08 PM Utility failed.
2012/02/28 14:42:20 PM Communication is lost.
2012/02/28 14:44:20 PM Communication is established.
```

**Figure 9: PowerPanel Server Log**

**Step 13** The client will then show the status has returned to normal.

```
[root@localhost etc]# /usr/sbin/pwrstat -status
The UPS information shows as following:
Current UPS status:
State..... Normal
Power Supply by..... Utility Power
Last Power Event..... Blackout at 2012/02/28 14:42:08
```

**Figure 10: Returned to Normal Status**

**Step 14** The configuration is completed.

# Serial Device Configuration on SRE-V



## Appendix A: Installing and Configuring Cyclades

- Step 1** Download the Cyclades 64 bit rpm file: **cyclades-serial-client-0.93-1.2.el5.rf.x86\_64.rpm** for CentOS 5. A couple of websites that provide this rpm are: <http://pkgs.org/download/cyclades-serial-client> and [http://rpmfind.net/linux/RPM/dag/redhat/el5/x86\\_64/cyclades-serial-client-0.93-1.2.el5.rf.x86\\_64.html](http://rpmfind.net/linux/RPM/dag/redhat/el5/x86_64/cyclades-serial-client-0.93-1.2.el5.rf.x86_64.html) . Otherwise please search for the rpm file online and download it.
- Step 2** Upload the Cyclades rpm file to the VM. For example, using curl the command format would be: “**curl -u <user> ftp://<ip>/<dir>/cyclades-serial-client-0.93-1.2.el5.rf.x86\_64.rpm>cyclades-serial-client-0.93-1.2.el5.rf.x86\_64.rpm**”.
- Step 3** As the root user install the rpm file via the command ‘**rpm -ivh cyclades-serial-client-0.93-1.2.el5.rf.x86\_64.rpm**’.
- Step 4** In the /usr/sbin/ directory there are two important Cyclades scripts: **cyclades-ser-cli** and **cyclades-serial-client**. Edit the file **cyclades-serial-client** to make the changes below.
- Replace all occurrences of **tsrsock** with **/usr/sbin/cyclades-ser-cli**
  - Replace the line: **opts=”-p 28672 \$opts**” with line: **opts=”-p 6000 \$opts**”
  - Save the file
- Step 5** Now go back to the section covering the specific serial device that you are configuring and continue following the steps.