User's Guide

PowerAlert® Network Shutdown Agent Software

Note: PowerAlert is not required to operate your UPS system. For the latest PowerAlert updates, go to tripplite.com/products/power-alert

1111 W. 35th Street, Chicago, IL 60609 USA • tripplite.com/support

1. Introduction

PowerAlert Network Shutdown Agent (PANSA) monitors one or two network-attached Tripp Lite UPS systems for specific power events, takes the appropriate action to gracefully shutdown operating systems or execute scripts, and protects systems from data loss or corruption. Example actions include shutting down the computer receiving UPS backup power when the UPS enters a low battery condition or, in a VMware application, signaling an ESXi Host to shut down. If the ESXi Host is a member of a high availability cluster, the desired response to the On Battery event could be to shut down the virtual machines running on the host before shutting down the host itself. This User's Guide will cover the basic scenarios.

Feature	Comments
Unattended computer operating system shutdown	Must monitor WEBCARDLX, SNMPWEBCARD or PowerAlert Local for shutdown signal
Autodiscovery of SNMP-enabled devices	Discovers devices accessible on the network via WEBCARDLX, SNMPWEBCARD, PowerAlert Local (software version 12.04.0040 and above) and PowerAlert Office software

Figure 1.1: PowerAlert Network Shutdown Agent Key Features

This document is divided into three sections, each covering implementation with a different operating system / environment. Please refer to the section that applies to your environment:

- **Section 2.** Windows Implementation
- **Section 3.** Linux Implementation
- **Section 4.** VMware Implementation

2. Windows Implementation

2.1 Pre-Installation

Note: PANSA requires pre-installation configuration of firewall software settings and/or WEBCARDLX/SNMPWEBCARD settings.

- Open UDP ports 161, 162, 3664 and 3665 on your computer's firewall software and disable the Windows SNMP trap service.
- The host computer for the device PANSA will be monitoring must also have UDP ports 161, 162, 3664 and 3665 open on its firewall software.
- If PANSA will be configured to monitor a WEBCARDLX, SNMPWEBCARD or PowerAlert Office software, enable SNMP access and define a read/write community string (the default community string for firmware version 12.04.0048 and above is **tripplite**).

Warning: Do not install more than one edition of a PowerAlert product on a single computer.

If you have a previous version of PANSA installed, Tripp Lite recommends you manually uninstall it prior to installing the new version.

2.2 Installation

Go to **tripplite.com/products/power-alert-network-shutdown-agent** to download the latest PANSA installer package for Windows.

Once downloaded, double-click the PowerAlert installer and follow the on-screen prompts. The installer will attempt to uninstall previous versions of PANSA (if this action was not performed during pre-installation).

If you experience issues installing PANSA, uninstall any current versions then right-click on the install file and select "Run as Administrator".

Current Windows platforms are supported.

2.3 Configuration

After installation, the option to launch the PANSA Console (Figure 2.3.1) becomes available. The Console contains all the controls required to operate PANSA.

The "Monitored UPS" fields will be blank when PANSA starts up for the first time. Click the "Discover Agents" button to open the discovery settings window (Figure 2.3.2). Enter an IP address and community name (**tripplite** is the default). You must click the "Add to list" button after each entry and can add multiple entries, if desired. Use the asterisk character (*) as a wildcard in the IP address field. For example, 192.168.1.* will cover 192.168.1.1 through 192.168.1.255. If you know the specific IP address of the device you
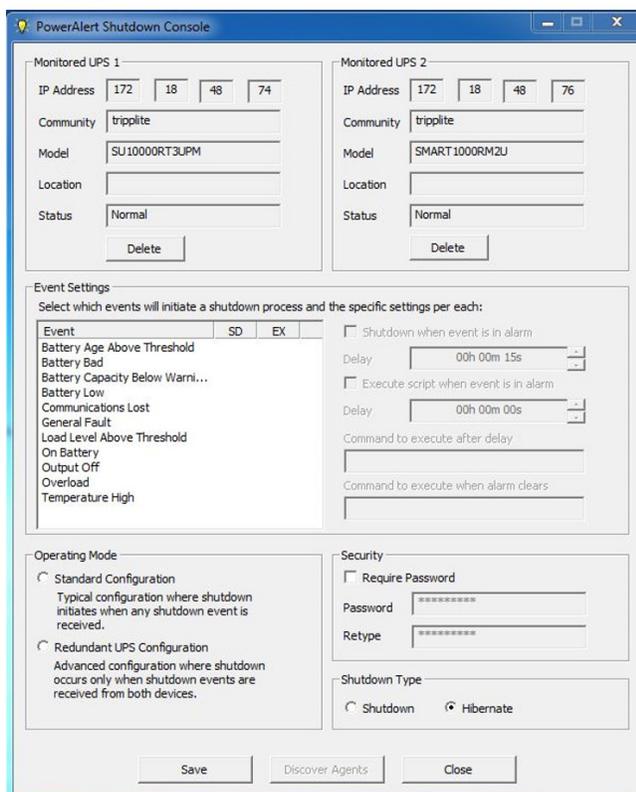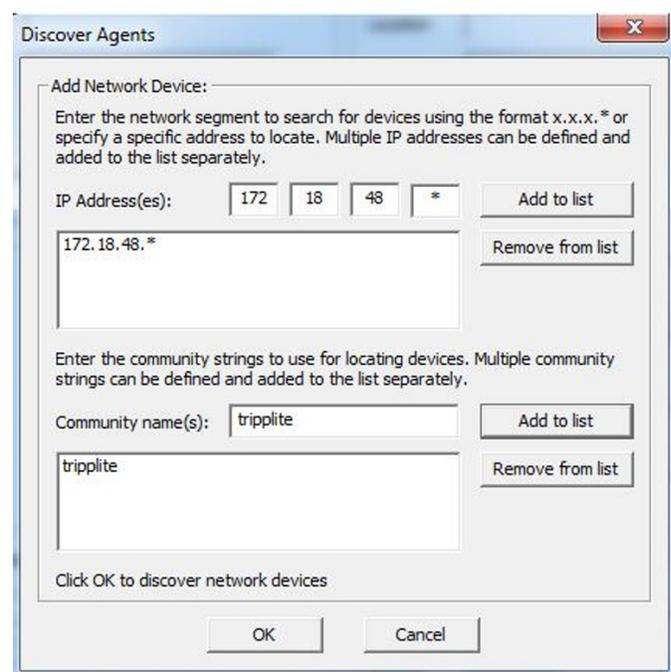

Figure 2.3.1

Figure 2.3.2

plan to monitor, enter that address. If you are unsure what information to enter in the discovery settings window, contact your network administrator for assistance.

2. Windows Implementation

Note: PANSA attempts to discover compatible WEBCARDLX and SNMPWEBCARD firmware, PowerAlert Local, PowerAlert Office software, but it does not search for PowerAlert Network Management System or other PowerAlert Network Shutdown Agents on the network.

After entering the IP addresses and community names to search, click "OK" to initiate the network autodiscovery process. Autodiscovery will detect SNMP-enabled Tripp Lite devices on the local network that PANSA is able to monitor. The results window (Figure 2.3.3) will list compatible devices detected by the autodiscovery process. If no devices are detected, confirm the devices have been configured with the firewall and other settings listed earlier.
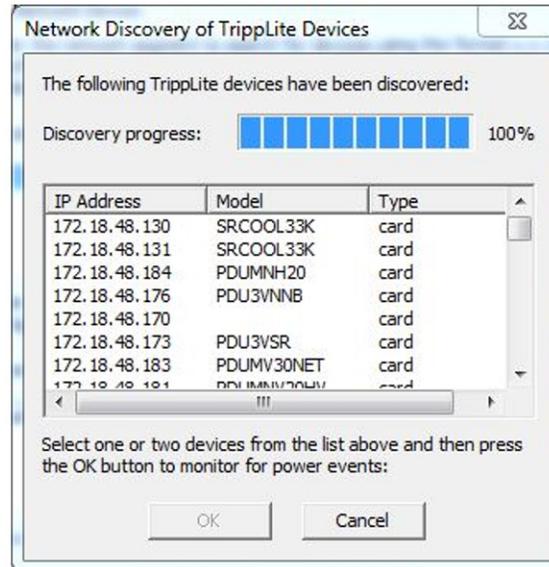

Figure 2.3.3

Note: Discovery errors will be logged in the C:\Program Files (x86)\TrippLite\PowerAlert\log directory in the PANMSdiscoveryerror.log file. This file can be viewed using a standard text editor such as Notepad.

Select either one or two systems to be monitored by PANSA, then click "OK" to return to the main console window. The device or devices selected will be displayed in the "Monitored UPS" fields. Each device's IP address, community name, model, location and status will be shown.

The Status field shows whether the monitored device is currently accessible. If the monitored device is offline, PANSA will attempt to reconnect to the monitored device continuously for 15 minutes, then once every 15 minutes. You can add devices by running the autodiscovery process again. Devices can also be deleted from the console, resulting in PANSA de-registering itself from the device.

2.3.1 Event Settings

After selecting one or two systems to monitor, you should define which monitored event(s) will prompt PANSA to either shut down your computer or execute a command. The list of events is dynamic, and includes items such as "Battery Low," "On Battery" and "Temperature High," as well as events that may be specific to the monitored device; supported events vary from device to device. For a complete list of events supported by a specific device, launch the PowerAlert Console for that device and examine the Events module.

Highlight one or more events in the list, then choose to enable shutdown and/or command execution when the selected event(s) is (are) detected. Events that are not selected will be ignored. Use the delay dials to set the amount of time PANSA will wait for the event to clear before shutting down the computer or executing a command.

2.3.2 Shutdown

When PANSA is triggered to act, it will perform either a shutdown or hibernation based upon the selected *Shutdown Type*. "Hibernate" will be an available choice only if the hibernation feature is supported by your computer's operating system. If it is supported, "Hibernate" will be the default setting.

2. Windows Implementation

2.3.3 Command Execution

PANSA manages and monitors the selected system(s) and can execute a command upon receiving a critical event, such as a power outage. It can also execute a command when the event clears, such as returning to utility power. The command can be used to launch an executable or to execute a batch file. Typically, the command execution is used to spawn a batch file that instructs other external systems to shut down, such as VMware servers. Refer to **Section 4** for more information on executing commands and scripts for VMware.

Warning! Any user with access to PANSA can change which script will be executed.

It is not recommended to use the Command Execution feature to run an executable file that interacts with the desktop. Windows Session 0 Isolation will prevent the Graphical User Interface (GUI) or popup called in the script from appearing on the Windows desktop and may lead to the script not executing in the same manner as when it is manually called. Examples of executables that interact with the desktop are: notepad.exe and putty.exe.

An example of batch file contents that would work as expected when manually executed but not when called using the Command Execution feature is:

C:\Windows\system32\notepad.exe

Manually executing a batch file with those contents would launch the Windows Notepad application, but calling the same file using the Command Execution feature would either lead to an Interactive Services Detection popup or no change on the Windows desktop, depending on the Windows Operating System in use. Users may see the notepad.exe process start in task manager, but the application will not launch.

It is recommended that scripts which require interaction with the desktop should be modified to remove this behavior. Examples of scripts that do not interact with the desktop are the VMware scripts found in the appendix of this manual.

An example of simple batch file content that will work when called through PANSA is:

whoami >> somefile.text

This file would print the name of the user executing the file into a file called "somefile.txt" in the current working directory. If "somefile.txt" is not in the current directory, it will be created. If "somefile.txt" does exist the name of the user executing the file will be appended to the file. If this file was called using PANSA, "somefile.txt" will be created in the "C:\Program Files (x86)\TrippLite\PowerAlert\engine" directory.

Tips for working with command execution:

• Enter the full path name (case sensitive) to the script, including the file extension, in the "Command to execute after delay" field (and if applicable, "Command to execute after alarm clears" field). Example: C:\Users\myuser\myflile.bat would be used to call "myfile.bat" from the "C:\Users\myuser" directory. The one exception to this rule is when the file is placed in the "C:\Program Files (x86)\TrippLite\PowerAlert\engine" directory, in which case only the file name needs to be entered. Short filenames (DOS 8.3) can be input into these fields if your system supports them.

• "pansa.exe" in the "C:\Program Files (x86)\TrippLite\PowerAlert\engine" directory will call the script when a configured event occurs.

• PANSA does not support spaces in the "Command to execute after delay" or "Command to execute after alarm clears" fields. If the path to your script contains spaces, consider placing the script in the "C:\Program Files (x86)\TrippLite\PowerAlert\engine" directory and simply inputting the filename and extension (preferred) or using DOS 8.3 filenames to call the script file.

2.3.4 Operating Mode

The "Operating Mode" setting applies only when PANSA is monitoring two UPS systems; it defines the conditions under which your computer will shut down and/or execute commands. If "Standard Configuration" is selected, your computer will shut down when either monitored device experiences a trigger event. If "Redundant UPS Configuration" is selected, your computer will shut down only when both monitored devices experience a trigger event.

Note: In a redundant UPS system configuration, each UPS system must have sufficient load capacity to handle the total power requirement of all connected equipment.

2.3.5 Security

The PANSA Console must authenticate with the PANSA service. By default, the password to perform this authentication is **tripplite**; PANSA uses this password by default to perform the requisite authentication. The "Security" setting determines whether a password other than **tripplite** is required by the PANSA Console to access the PANSA service.

2.4 System Tray Icon

The PANSA status icon is located in the Windows system tray and provides a convenient visual notification of power status. During normal conditions, a light bulb icon will display, as shown in Figure 2.4.1. If you place the mouse pointer over the icon, a tooltip will display text to explain the icon's meaning. Right-clicking on the icon will show a popup menu listing available options.

Figure 2.4.1

The status icon and tooltip text will change to indicate corresponding changes in power and UPS system status. The possible icon representations are as follows:

Note: Icon graphics may differ based on the operating system of the PANSA host computer.

The light bulb icon is displayed when all devices are on utility power and running under normal conditions.

The yellow warning icon is displayed when one or more warning-level events—monitored or not—are triggered. Example: a UPS running on battery power.

The red "X" icon is displayed when one or more critical-level events—monitored or not—are triggered. Example: a UPS running on battery with a low-battery condition.

The black "X" icon is displayed when the PANSA Console cannot communicate with the target device. This indicates that either the UPS is offline or a network error is restricting communications.

The lock icon is displayed when security is enabled and the password is changed from the default password **tripplite**. When locked, the PANSA Console user must enter the correct password, which enables the Console to communicate with the PANSA service.

2.5 Performing a Silent Installation of PowerAlert

This procedure is applicable to version 12.4.58 (and later) of PANSA for Windows.

1. Using the computer on which PowerAlert is to be installed, create a folder and place the PANSA installation file in it.

2. Open a command prompt window and change the path location to the folder created in the previous step.

    Note: On many Windows operating systems, an elevated command prompt will be required for this step.

3. Execute the following command*:

    <<filename>>**.exe /S /v/qn**

    Note: <<filename>>.exe is an example. The actual PANSA installation file name should be entered in its place.

    The PANSA installation will complete without displaying any installation windows or questions.

    *For legacy versions of PANSA for Windows, the command to be used in step 3 is:
    <<filename>>**.exe /s /v" /qn FILELOCATION=C:**<<folder name>>"

    Note: <<filename>>.exe is an example. The actual PANSA installation file name should be used in its place.

    Note: C:\<<folder name>>"is an example. The actual folder name containing the PANSA installation file should be used in its place.

2.6 Using the paconfig.ini File

PANSA can be configured using a configuration file named "paconfig.ini". Versions 12.04.0058 and later of PANSA for Windows and versions 12.04.0057 and later of PANSA for Linux include example paconfig.ini files in the "poweralert" directory upon install.

The paconfig.ini file can be viewed and edited in a text editor, like Notepad (Windows), or vi (Linux). PANSA will use the settings defined in a properly configured "paconfig.ini" file if the file is placed in the "PowerAlert" directory and PowerAlert is then reset. See **Section 5** for instructions on resetting PANSA.

3. Linux Implementation

3.1 Pre-Installation

Note: PANSA requires pre-installation configuration of firewall software settings and/or WEBCARDLX/SNMPWEBCARD settings.

- Open UDP ports 161, 162, 3664 and 3665 on your computer's firewall software.
- The host computer for the device PANSA will be monitoring must also have UDP ports 161, 162, 3664 and 3665 open on its firewall software.
- If PANSA will be configured to monitor a WEBCARDLX or SNMPWEBCARD, enable the card's SNMP access and define a read/write community string (the default community string for firmware version 12.04.0048 and above is **tripplite**).

Warning: Do not install more than one edition of a PowerAlert product on a single computer.

If a previous version of PANSA has been installed, Tripp Lite recommends you manually uninstall it prior to installing the new version.

3.2 Installation

1. Go to **tripplite.com/products/power-alert-network-shutdown-agent** to download the latest PANSA installer package for the specific Linux operating system.
2. Copy the appropriate software file to your local machine.
3. Open a terminal used for your local machine.
4. Change to the directory where the file is located and enter the following command: **sudo rpm -i <rpm-file-name> [--nodeps]**. Use the RPM main page for assistance with any installation issues.

PANSA will install to the directory **/var/tripplite/poweralert** and will place or install the daemon process for each software version within **/var/tripplite/poweralert/engine**.

3.3 Configuration

After installation, PANSA can be configured by modifying its configuration file.

3.3.1 Configuring PANSA Using the Configuration File

The PANSA installation package includes a sample configuration file, which can be used to configure the system without a graphical interface (GUI). All settings available in the GUI Console are available in the configuration file. Read **Section 2.3** prior to modifying a configuration file.

Below is content from the sample configuration file included in the installation package:

```
[PA_Remote]
Server=10.15.0.8
SNMPMacAddress=00:06:67:23:66:6d
SNMPVersion=2
SNMPCommunity=tripplite
SNMPPort=161
SNMPTrapPort=162
```

```
[PA_Remote_2]
Server=10.15.0.9
SNMPMacAddress=00:06:67:23:66:6e
SNMPVersion=2
SNMPCommunity=tripplite
SNMPPort=161
SNMPTrapPort=162
```

```
[RemoteEvent:On Battery]
EnableShutdown=true
ShutdownDelay=15
EnableExecuteCommand=false
ExecuteCommandDelay=5
ExecuteCommandOnSet=/home/vi-admin/shutdownesxi.sh
ExecuteCommandOnClear=
```

```
[PA_Engine]
ShutdownRequired=All
```

3. Linux Implementation

The configuration file must be modified and configured for use in your environment as follows:

1. Change the directory to the root installation directory: **cd /var/tripplite/poweralert** .

2. Copy the sample configuration file: **sudo cp example_paconfig.ini paconfig.ini** .

3. Edit the configuration file: **sudo vi paconfig.ini** .

4. The [PA_Remote] section identifies the attributes of the primary remote system being monitored.

   a. The Server keyword is the remote system's IP address. Change this value to the IPv4 address of the remote system.

   b. SNMPMacAddress is the MAC address of the remote system identified in 4a, above. This keyword is optional and can be omitted if not known. If omitted, the entire line must be removed from the configuration file.

   c. SNMPVersion specifies the SNMP version in use by the remote system. This value should correspond with the credentials configured on the WEBCARDLX, SNMPWEBCARD or PowerAlert Local system. The valid values are: 1 for SNMPv1, 2 for SNMPv2c. SNMPv3 is currently not supported by PANSA.

   d. SNMPCommunity is the community string used to authenticate with the remote system. The default value is **tripplite** and may only be changed if the same read/write community has been defined on the remote system. Remote PowerAlert Local systems require a community string of **tripplite**.

   e. SNMPPort is the port the remote system listens for SNMP-get and SNMP-set requests. By default, WEBCARDLXs and SNMPWEBCARDs are configured for port 161 and PowerAlert Local systems are configured for port 3664. The value should correspond with the remote system.

   f. SNMPTrapPort is the port the remote system sends SNMP traps to PANSA. By default, WEBCARDLXs and SNMPWEBCARDs are configured for port 162 and PowerAlert Local systems are configured for port 3664. The value should correspond with the remote system.

5. The [PA_Remote_2] section identifies the attributes of the secondary remote system being monitored.

   a. **If monitoring one UPS system:**

      i. Remove the entire [PA_Remote_2] section from the configuration file.

      ii. In the [PA_Engine] section, remove ShutdownRequired, as it is not needed for a single UPS configuration.

      iii. Skip to Step 6 below.

   b. **If monitoring two UPS systems:**

      i. Using the same instructions in Step 4, make the necessary modifications as they apply to the secondary system.

      ii. If the two systems are configured for redundancy, then in the [PA_Engine] section, keep the ShutdownRequired=All value. This will cause the shutdown and/or command execution to occur when both UPS systems are in an alarm condition. If the two UPS systems are configured independently, change the value to ShutdownRequired=One. This will cause the shutdown and/or command execution to occur when either UPS system is in an alarm condition.

6. Each [RemoteEvent:<EventName>] section in the configuration file describes the options for each event supported by the selected remote system(s). The file contents described at the beginning of this section are for an "On Battery" event. This is the event that triggers when a power outage occurs and the UPS goes on battery. If you require maximum runtime, you can change the section name to [RemoteEvent:Low Battery]. If you would like one delay or command to execute when on battery, but different settings for low battery, you can make a copy of the section so both events are listed in the configuration file.

   a. The EnableShutdown keyword defines whether or not the event is selected. If the value is "true", then the event is selected and PANSA will monitor the event. If "false", the event will not be used to trigger a system shutdown.

   b. ShutdownDelay specifies the number of seconds PANSA will wait for the event to clear before shutting down the computer when and if EnableShutdown is set to "true".

   c. The EnableExecuteCommand keyword defines whether or not the event is selected. If the value is "true", then the event is selected and PANSA will monitor the event. If "false", the event will not be used to trigger a command execution.

   d. ExecuteCommandDelay specifies the number of seconds PANSA will wait for the event to clear before executing the command specified for ExecuteCommandOnSet, when and if EnableExecuteCommand is set to "true". This value does not apply to ExecuteCommandOnClear, which is executed immediately when and if EnableExecuteCommand is set to "true" and the event clears.

   e. The value specified for ExecuteCommandOnSet is the command that will execute when the delay expires after the event is triggered. This value is optional. If not specified, no command will execute when the event is triggered.

   f. The value specified for ExecuteCommandOnClear is the command that will execute immediately when the event clears. The value is optional. If not specified, no command will execute when the event clears.

   Note: If neither EnableShutdown nor EnableExecuteCommand is set to "true", the event will be omitted and will not be monitored.

7. The [PA_Engine] section specifies system options for PANSA itself.

    a. ShutdownRequired specifies whether shutdown and/or command execution will occur when an event is triggered on one or both UPS systems. If set to "All", an event must occur on both UPS systems before shutting down or executing a command. If set to "One", an event must occur on only one UPS. Refer to **Section 2.3** for related information.

8. Save the changes to the configuration file: **:wq<CR>** (CR=carriage return).

After the configuration file is changed, you must stop PANSA, delete all data files and restart it. Do this by entering the following commands:

sudo service pansad stop
cd /var/tripplite/poweralert/data
sudo rm –rf *
sudo service pansad start

It is also important to ensure PANSA is configured to run at startup. Enter the following command to enable running PANSA at startup:

sudo chkconfig pasdad on

3.4 Uninstalling PANSA

To uninstall PANSA type the following command:

sudo rpm -e <rpm-file-name>

where <rpm-file-name> is the name of the installed PANSA RPM. For example: sudo rpm -e pansa-12.04.0059-openSUSE11_4x86_64

If the sudo rpm -e <rpm-file-name> command does not work, try:

sudo rpm -e pansa

3.5 Using the poweralert.conf File

The poweralert.conf file can be used to configure certain functionality of PANSA, including the syslog level, the location and logging level of the poweralert.log file, and the type of shutdown command when a system shutdown is sent. The poweralert.conf file is included with PANSA versions 12.04.0057 and later for Linux operating systems; it can be edited using a text editor (e.g. vi).

PANSA will use the settings in a properly configured poweralert.conf file in the /var/tripplite/poweralert directory after the PANSA service is restarted. To restart the PANSA service on most supported Linux operating systems, issue the command:

sudo service <service_name> restart

The <service_name> value should be pasdad (PANSA version 12.04.0057) or pansad (versions later than 12.04.0057).

A common use case for the poweralert.conf file is to assist in troubleshooting by changing the filelog.level value to TRACE or DEBUG.

For reference, see **Appendix B** for the contents of the poweralert.conf file.

4. VMware Implementation (ESXi Server Environments)

This section covers PANSA usage with ESXi and vCenter 6.7 and Later. Refer to the Appendices for information regarding PANSA usage with earlier versions.

4.1 PANSA for ESXi and vCenter 6.7 and Later

PANSA 12.05 for Windows can monitor a compatible UPS or pair of UPS systems, and shutdown ESXi 6.7* or later hosts based on UPS events such as On Battery. There are two standard implementation options; the sections that follow describe each:

- Graceful shutdown of ESXi hosts NOT in an HA (high-availability) cluster
- Graceful shutdown of ESXi hosts in an HA cluster

Free versions of ESXi are not compatible with this solution due to a known issue where the shutdown scripts are not allowed to write to the servers and initiate shutdown.

4.1.1 Graceful Shutdown of ESXi Hosts NOT in an HA Cluster (Standard Configuration)

This section details the configuration required if the ESXI hosts are NOT in an HA Cluster.
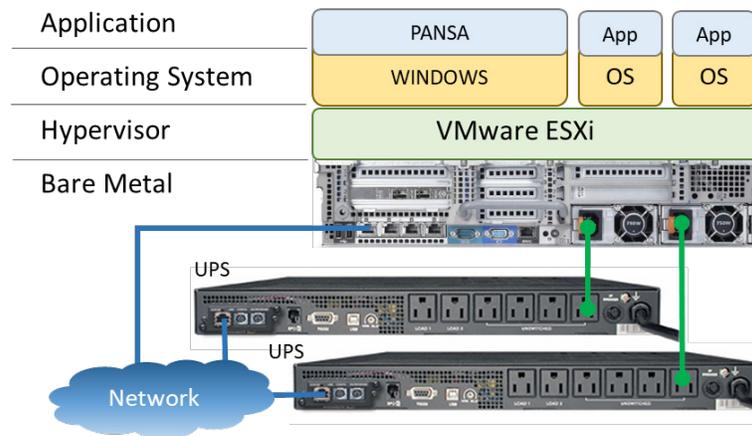
Prerequisites

- ESXi 6.7-7.x with a purchased license
- VMware tools installed on the virtual machines being gracefully shutdown
- A Windows machine running PANSA with network access to the ESXi hosts. This can be a virtual machine on the host or any other physical computer/virtual machine.

Configure Virtual Machine Startup/Shutdown

The ESXi hosts must be configured to automatically shut down guest operating systems. For guest operating systems to turn back on when the ESXi hosts turn on, also configure the autostart/startup behavior.

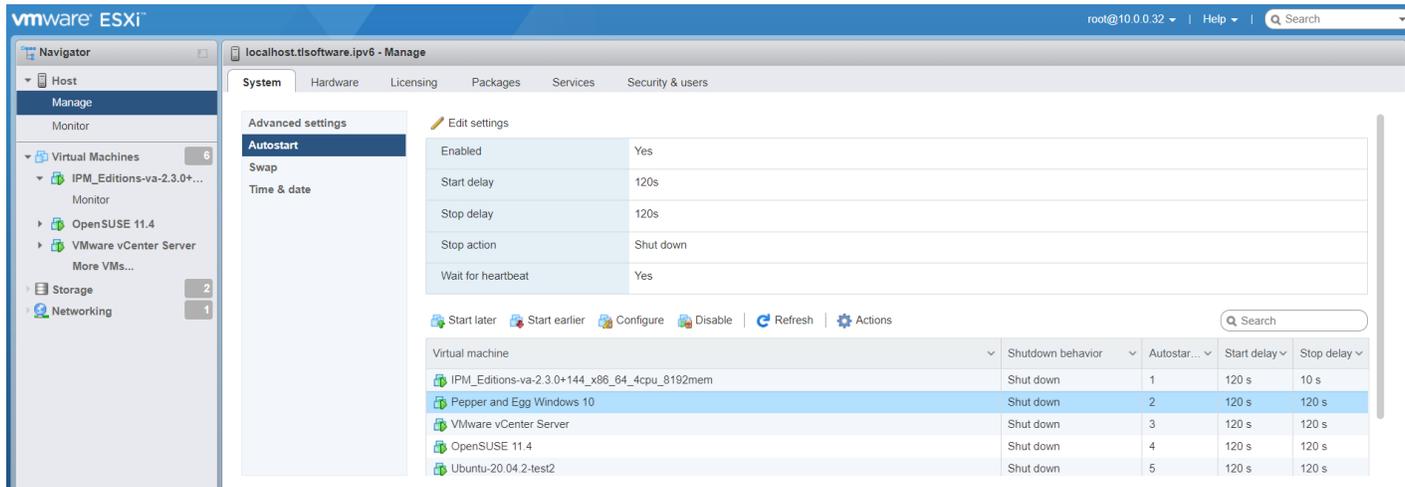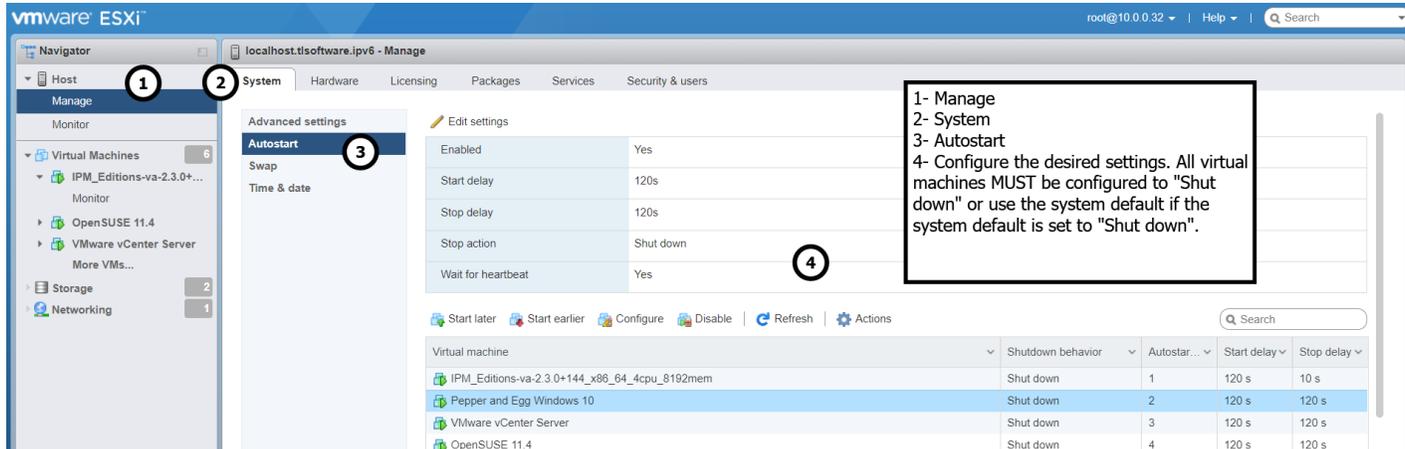# 4. VMware Implementation (ESXi Server Environments)

Configure Virtual Machine Startup/Shutdown from the GUI of an ESXi Host

To configure virtual machine startup/shutdown from the graphical user interface (GUI) of an ESXi host, perform the following steps:

1. Click "Host" → "Manage"

2. Click "System"

3. Click "Autostart"

4. Configure the "Shutdown behavior" for all virtual machines to "Shut down"

   Optional: Configure the "Autostart" behavior and order to ensure the virtual machines automatically restart when the ESXi host turns on

5. Repeat 1-4 for all ESXi hosts

Note: To alternatively enable shutdown for all virtual machines, set the system "Stop" action to "Shut down" and set the "Shutdown Behavior" of the virtual machines to "System Default".

4. VMware Implementation (ESXi Server Environments)

Configure Virtual Machine Startup/Shutdown from the vCenter Client

If vCenter is being used to manage ESXi hosts, configure startup/shutdown settings in vCenter by performing the following steps:

1. From the vCenter Client, click the ESXi host

2. Click "Configure"

3. Click "VM Startup/Shutdown"

4. Edit settings to enable "Guest Shutdown" for all the virtual machines

   Optional: Configure the "Startup behavior" and "Startup Delay(s)" to ensure the virtual machines automatically restart when the ESXi host turns on.

5. Repeat 1-4 for all ESXi hosts

For more information on configuring Virtual Machine Startup and Shutdown options, review the VMware documentation at the following link:

https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.vm_admin.doc/GUID-7A132146-2779-4F4B-B39D-EE4F129DB79C.html

Install PowerCLI

PowerCLI must be installed on the computer running PANSA; several installation methods are available. One method is to run PowerShell as an administrator, then execute the following command:

Install-Module -Name VMware.PowerCLI

For more information on the PowerCLI see:

https://developer.vmware.com/web/tool/12.4/vmware-powercli

4. VMware Implementation (ESXi Server Environments)

Allow Remote Signed PowerShell Scripts (Optional)

By default, the .bat script PANSA uses to call the shutdown script uses the "PowerShell.exe -ExecutionPolicy ByPass" command, so PowerShell script execution policies should not prevent it from running. However, the PowerShell scripts are signed and advanced users may wish to remove the -ExecutionPolicy ByPass parameter from the .bat file or run the PowerShell scripts directly. Without the "-ExecutionPolicy ByPass" parameter, for PANSA to run the shutdown script, the Windows machine must have a PowerShell execution policy that is set to RemoteSigned (or a less restrictive policy) for the specific script.

Check current script execution settings by running the following PowerShell command:

Get-ExecutionPolicy

```
Administrator: Windows PowerShell
PS C:\Users\devops> Get-ExecutionPolicy
Restricted
PS C:\Users\devops>
```

To set ExecutionPolicy for allowing RemoteSigned scripts, run PowerShell as an administrator, then execute the following command:

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned

```
Administrator: Windows PowerShell
PS C:\Users\devops> Set-ExecutionPolicy -ExecutionPolicy RemoteSigned
Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security
risks described
in the about_Execution_Policies help topic at https:/go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): y
PS C:\Users\devops>
```

To configure more granular PowerShell script execution policies, review the following Microsoft document:

https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_execution_policies

Configure access.conf

1. Open a text editor (e.g. Notepad) and navigate to the access.conf file in the C:\Program Files (x86)\TrippLite\PowerAlert\VMware_ scripts\6.7andnewer\Not_HA directory.
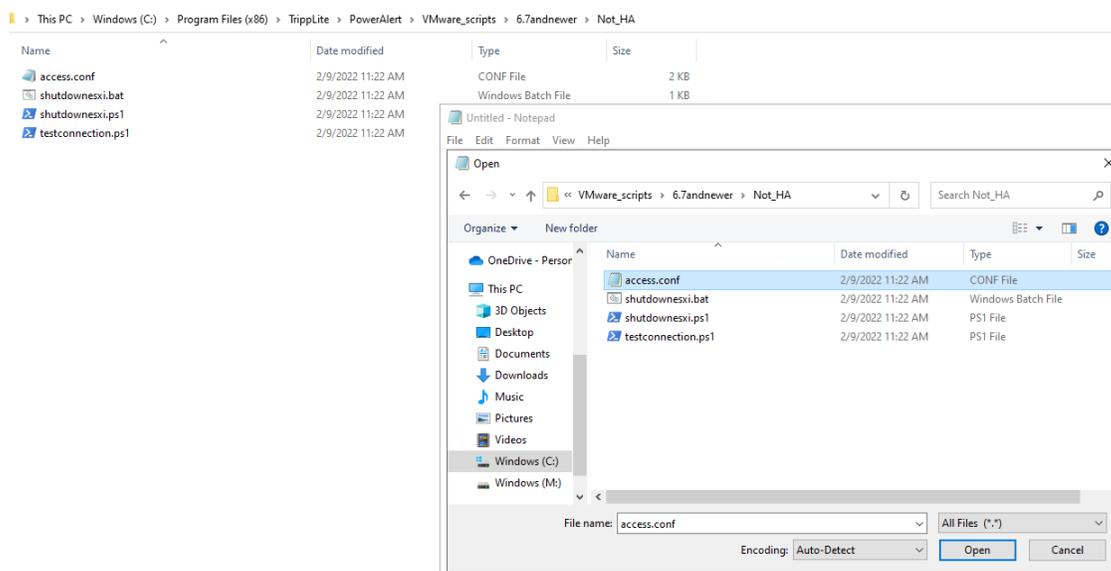
Note: Administrative privileges may be required to save the files in the C:\Program Files (x86)\TrippLite\PowerAlert\VMware_scripts\6.7andnewer\Not_ HA directory. Run text editor as administrator, if required.

13

4. VMware Implementation (ESXi Server Environments)
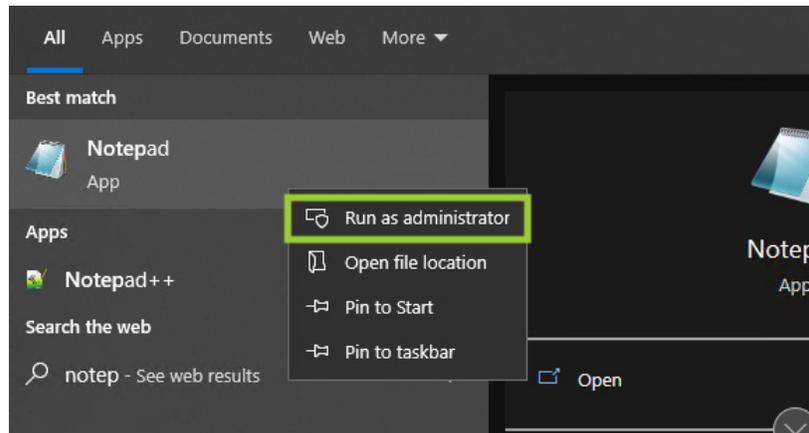
2. Change the hostslist, username, and password values

   a. Hostslists - A list of IP addresses or Fully Qualified Domain Names (FQDNs) of all the ESXi hosts to be shut down. Addresses should be separated by a comma, no space.

   b. Username - A username for an ESXi account that can shut down the hosts

   c. Password - The password for the ESXi username

   Note: The other access.conf values are NOT used in the Standard Case; retain their default settings.

3. Save the changes. See Appendix C for examples.

Move Files to the "Engine" Directory

1. Copy all four files from C:\Program Files (x86)\TrippLite\PowerAlert\VMware_scripts\6.7andnewer\Not_HA to C:\Program Files (x86)\TrippLite\PowerAlert\engine\
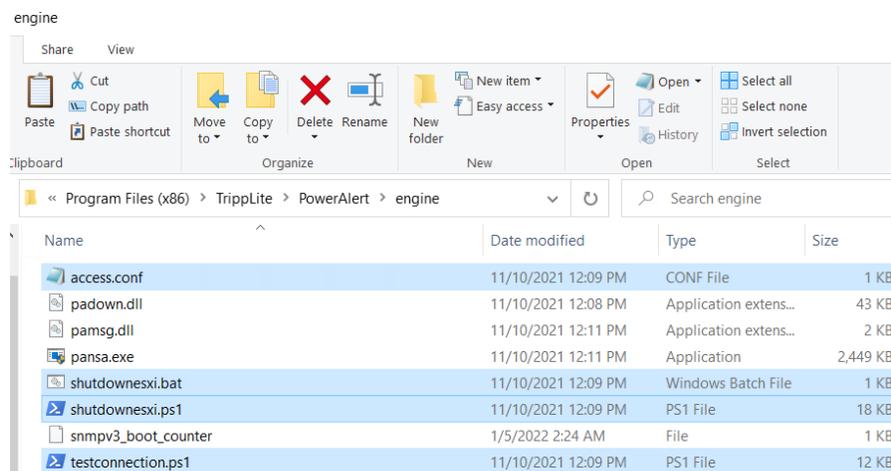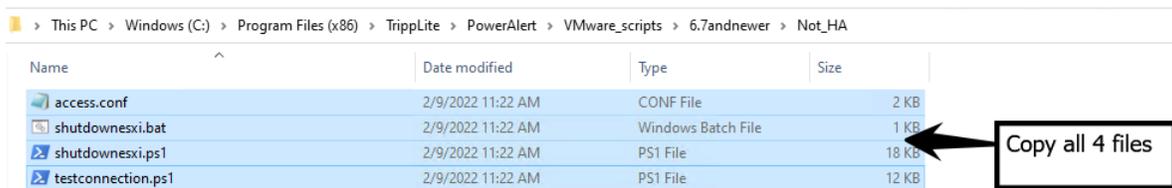
Paste them into the "C:\Program Files (x86)\TrippLite\PowerAlert\engine" directory

14

4. VMware Implementation (ESXi Server Environments)

2. For the event triggering the shutdown, check the "Execute script when event is in alarm" box.

3. Configure a "Delay" for the shutdown script execution. If the shutdown event clears before this delay elapses, the shutdown will not occur.

4. VMware Implementation (ESXi Server Environments)

4. Type shutdownesxi.bat in the "Command to execute after delay" field.

5. [Optional] Repeat steps 2-4 for additional trigger events, as needed.
6. Click "Save".

PANSA is now configured to execute the shutdown script that will gracefully shut down the specified ESXi hosts.

4. VMware Implementation (ESXi Server Environments)

4.1.2 Graceful Shutdown of ESXi Hosts in an HA Cluster (Standard Configuration)

This section details the configuration required if ESXi hosts are in an HA cluster.
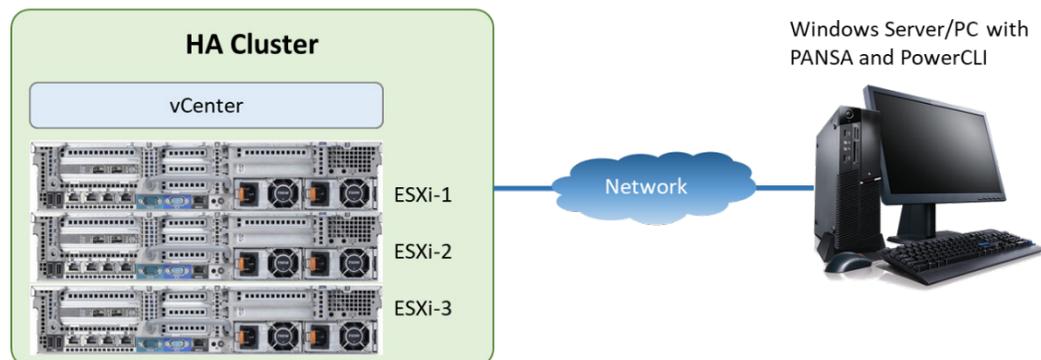
Prerequisites

- ESXi 6.7-7.x and vCenter 6.7-7.x with a purchased license

- VMware tools installed on the virtual machines being gracefully shutdown

- Windows machine running PANSA that is outside of the cluster with network access to vCenter and the ESXi hosts

Install PowerCLI

PowerCLI must be installed on the computer running PANSA; several installation methods are available. One method is to run PowerShell as an administrator, then execute the following command:

Install-Module -Name VMware.PowerCLI

For more information on the PowerCLI, see:

https://developer.vmware.com/web/tool/12.4/vmware-powercli

Allow Remote Signed PowerShell Scripts (Optional)

By default, the .bat script PANSA uses to call the shutdown script uses the "PowerShell.exe -ExecutionPolicy ByPass" command, so PowerShell script execution policies should not prevent it from running. However, the PowerShell scripts are signed and advanced users may wish to remove the -ExecutionPolicy ByPass parameter from the .bat file or run the PowerShell scripts directly. Without the "-ExecutionPolicy ByPass" parameter, for PANSA to run the shutdown script, the Windows machine must have a PowerShell execution policy that is set to RemoteSigned (or a less restrictive policy) for the specific script.

Check current script execution settings by running the following PowerShell command:

Get-ExecutionPolicy

To set your ExecutionPolicy to allow RemoteSigned scripts, run PowerShell as an administrator and execute this command:

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned

To configure more granular PowerShell script execution policies, see the following Microsoft document:

https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_execution_policies

17

4. VMware Implementation (ESXi Server Environments)

Configure access.conf

1. Open a text editor and navigate to the access.conf file in the C:\Program Files (x86)\TrippLite\PowerAlert\VMware_ scripts\6.7andnewer\HA_cluster directory.

Note: Administrative privileges may be required to save the files in the C:\Program Files (x86)\TrippLite\PowerAlert\VMware_scripts\6.7andnewer\HA_ cluster directory. Run text editor as administrator, if required.
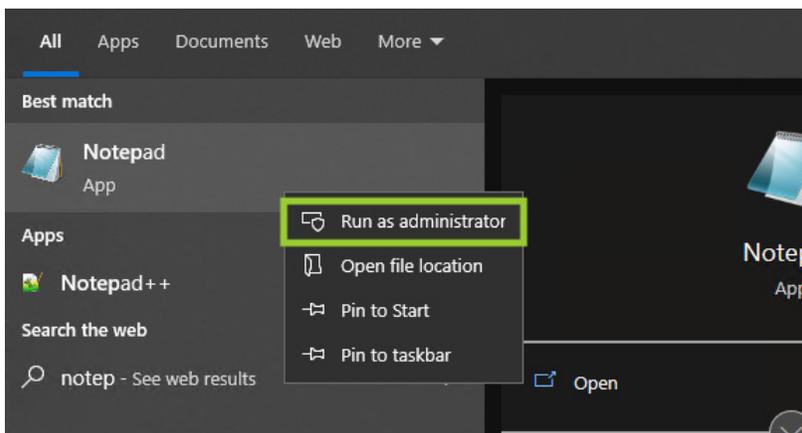
2. Change the following values in access.conf to match the environment's settings:

    a. vcenter- The IP address or Fully Qualified Domain Names (FQDN) of the vCenter installation that manages the cluster.

    b. vcentername- If vCenter is running as a virtual machine, input the name of the virtual machine. In most cases, the vcentername value is VMware vCenter Server. The image below demonstrates where vcentername can be validated in the vSphere Client.

18
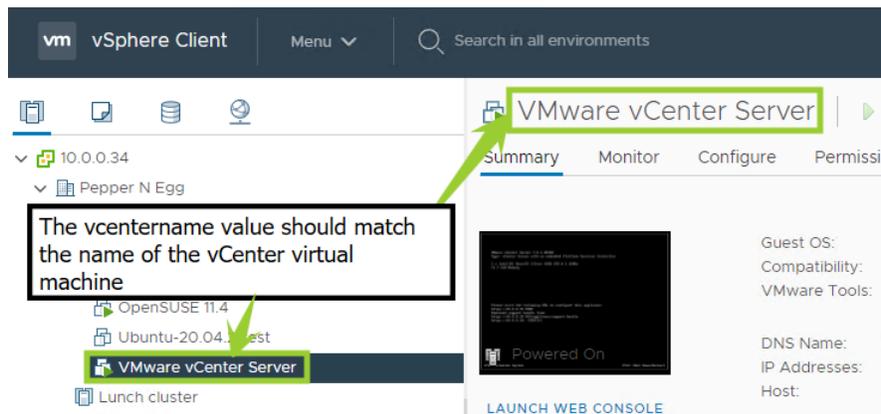
c. vcentershutdown- If vCenter is running on an ESXi host that will be shutdown, leave the value set to true. If vCenter is running on a hypervisor outside of the cluster that will be shutdown, set the value to false.

d. vcenterusername- A username for a vCenter user that can perform shut down of virtual machines in the cluster.

e. vcenterpassword- The password for the vcenterusername account.

f. hostslists- A list of IP addresses or Fully Qualified Domain Names (FQDNs) of all the ESXi hosts to be shut down. Addresses should be separated by a comma, no space.

g. username- The username of an ESXi account that can shut down the hosts.

h. password- The password for the ESXi username.

i. vmtimeout- The time (in seconds) that the shutdown script will wait before forcibly shutting down unresponsive virtual machines other than the vcenter virtual machine. The default value is 240 (4 minutes).

j. vctimeout- The time (in seconds) that the shutdown script will wait before forcibly shutting down the vcenter virtual machine. The default value is 120 (2 minutes).

The upsvmname is NOT used and should be left blank for standard configurations.

3. Save the changes. See Appendix C for examples.

Move Files to the "Engine" Directory

Copy all four files from C:\Program Files (x86)\TrippLite\PowerAlert\VMware_scripts\6.7andnewer\HA_cluster to C:\Program Files (x86)\TrippLite\PowerAlert\engine\

2. For the event triggering shutdown, check the "Execute script when event is in alarm" box.

3. Configure a "Delay" for the shutdown script execution. If the shut down event clears before this delay elapses, the shut down will not occur.

4. Type shutdownvcenter.bat in the "Command to execute after delay" field.

5. [Optional] Repeat steps 2-4 for additional trigger events s as needed.
6. Click "Save".

PANSA is now configured to execute the shutdown script that will gracefully shut down the specified HA cluster.

5. Troubleshooting

Problem	Possible Solutions
PowerAlert Network Shutdown Agent is not able to discover the desired device.	Open UDP ports 161, 162, 3664 and 3665 on your computer's firewall software and disable the Windows SNMP trap service. The device's host computer that PowerAlert Network Shutdown Agent will be monitoring must also have UDP ports 161, 162, 3664 and 3665 open on its firewall software. If PowerAlert Network Shutdown Agent will be configured to monitor a WEBCARDLX or SNMPWEBCARD, enable the card's SNMP access and define a read/write community string (**tripplite** is the default for firmware version 12.04.0048 and above).
	The computer on which PowerAlert Network Shutdown Agent is installed should have a static IP address. If the address is dynamic, it is recommended to change this to a static address.
	Make sure all computers and devices are on the same network subnet and confirm that the TCP/IP network connections and local device communications are optional.
	The monitored PowerAlert software or SNMPWEBCARD firmware must be version 12.04.0040 or above.
	Check the "poweralert.log" and "PANMSDiscoveryError.log" files at "C:\Program Files (x86)\TrippLite\PowerAlert\log" (Windows) or "/var/tripplite/poweralert/log". These files can be opened with a standard text editor like Notepad or vi.
PowerAlert Network Shutdown Agent detects the device, but the device status does not change when a customer configured event occurs.	Open UDP ports 161, 162, 3664 and 3665 on your PANSA computer's firewall software and disable the Windows SNMP trap service.
	If the target is PowerAlert Local software or a 12.04.0055 (or earlier) version SNMPWEBCARD, confirm that the community string has been changed from "tlremote" to "tripplite"; in the GUI, go to Settings > Contacts > SNMP.
vMA error message similar to "Cannot find the specified target in vMA configuration. Error connecting to server at 'https://localhost/sdk/webService'" when attempting to run the shutdownvms.sh script.	Execute the command **vifp addservers <ESXi_host_IP_or_FQDN>** and enter the credentials for the ESXi hosts, then attempt to run the script again. Input the actual ESXi host IP address or Fully Qualified Domain Name in place of **<ESXi_host_IP_or_FQDN>** .
Issues inputting the path to "C:\Program Files (x86)\VMware\VMware vSphere CLI\bin\shutdownvms.pl" in the PANSA GUI.	Try using the short filename (DOS 8.3) for the path if your system supports it.
	Create a batch file that calls the script, place the batch file in the "C:\Program Files (x86)\TrippLite\PowerAlert\engine" directory. Simply type the name of the batch file into the "Command to execute after delay" field. Example batch file contents:
	@echo off
	REM remove (x86) from the path on 32 bit systems: cd "C:\Program Files (x86)\VMware\VMware vSphere CLI\bin" shutdownvms.pl
Instructions for Resetting PANSA 12.04.0055 And Later (Windows)	Many issues with PANSA can be resolved by resetting the software as described in the process below. Please note this process will erase any previous data and, unless a "paconfig.ini" file with alternate settings is used, will reset PowerAlert to default values.
	1. If the PowerAlert console (graphical user interface) is open, close it.
	2. Stop the PowerAlert Agent service in the Windows Services menu.
	Note: The Windows Services menu can be accessed in multiple ways, including navigating to "Control Panel\All Control Panel Items\Administrative Tools" or entering "services.msc" into a Windows Run prompt.
	3. Delete all files in the "C:\Program Files (x86)\TrippLite\PowerAlert\data" directory.
	4. Start the PowerAlert Agent service in the Windows Services menu.
	5. Wait 30 seconds and attempt to use PANSA.

5. Troubleshooting

Problem	Possible Solutions
Instructions for Resetting PANSA 12.04.0057 And Later (Linux)	Please note this process will erase any previous data. Unless a "paconfig.ini" file with alternate settings is used, this will reset PANSA to default values. All versions of PANSA 12.04.0057 and later (at the time of this User's Guide release) are "headless" and configured using the command line interface (CLI) of the Linux operating system they are installed on. Examples of the commands required to execute steps 1-3 are shown below.
	1. Stop the PANSA service. For PANSA versions prior to 12.04.0059: user@examplecomputer:~> sudo service pasdad stop For PANSA versions 12.04.0059 and later: user@examplecomputer:~> sudo service pansad stop
	2. Delete all the files in the "var/tripplite/poweralert/data" directory. user@examplecomputer:~> cd /var/tripplite/poweralert/data/ user@examplecomputer:/var/tripplite/poweralert/data/> sudo rm *
	3. Start the PANSA service. For PANSA versions prior to 12.04.0059: user@examplecomputer:/var/tripplite/poweralert> sudo service pasdad start
	For PANSA versions 12.04.0059 and later: user@examplecomputer:/var/tripplite/poweralert> sudo service pansad start
	4. Wait 30 seconds and attempt to use PANSA.

6. Technical Support

Before contacting Tripp Lite Technical Support, refer to **Section 5**. for possible solutions. If you are still unable to resolve the problem, contact Tripp Lite Technical Support at:

E-mail: techsupport@tripplite.com

Web: The latest PowerAlert software updates are available at **tripplite.com/software/**

Technical Support Assistance: tripplite.com/support

APPENDIX A – PANSA for VMware ESXi 5.x and 6.x

PANSA is compatible with VMware ESXi 5.x and ESXi 6.x* servers and can be installed on a virtual machine, or physical machine with the required network access, to monitor the UPS and trigger an ESXi shutdown upon detecting an event, such as a power outage. There are three implementation choices:

A.1 PANSA for VMware vSphere Management Assistant (vMA)

Note: *vMA is not compatible with ESXi, beginning with version 6.7.*

A.2 PANSA for standard Operating Systems

A.3 PowerAlert Local (Use Windows or a supported Linux operating system to configure ESXi shutdown)

Items A.1 and A.2 require a Tripp Lite WEBCARDLX or SNMPWEBCARD installed on the UPS powering the ESXi Server.

* Free versions of VMware ESXi Server are incompatible with PANSA for VMware due to a known issue in the free version of VMware, wherein PANSA shutdown scripts are not permitted to write to the server and initiate shutdown. Purchasing a licensed version of ESXi Server will resolve this issue.

To help determine which solution is the best fit for your application, review the flow chart below.

Are your ESXi hosts in a High-Availability (HA) cluster?

NO → **Which do you prefer to use?**

- **Windows** machine (physical or virtual)
  - Install the VMware vSphere CLI** and PANSA for Windows on the Windows machine
  - Use the shutdownesxi.bat script

Both require network access to the hosts

- **vMA**
  - Install PANSA for Linux/OpenSUSE/vMA on the vMA.
  - Use the shutdownesxi.sh script

YES → **Which do you prefer to use?**

- **Windows** machine (physical or virtual)
  - Install the VMware vSphere CLI** and PANSA for Windows on the Windows machine that is NOT in the same cluster as the hosts
  - Use the shutdownvms.pl script

Both require network access to the hosts and must NOT be in the same cluster as the hosts

- **vMA**
  - Install PANSA for Linux/OpenSUSE/vMA on the vMA
  - Use the "vifp addserver" command to add the hosts to the vMA
  - Use the shutdownvms.sh script

**The vSphere CLI 6.5 and later require the user to install a compatible version of Perl (e.g. ActivePerl or Strawberry Perl version 5.14).

A.1 PANSA for VMware vSphere Management Assistant (vMA)

One implementation of PANSA for VMware is to install VMware's vSphere Management Assistant (vMA) on a virtual machine on the ESXi server and then install PANSA onto the vMA. This is depicted in figure A.1.1 below.

Figure A.1.1

24
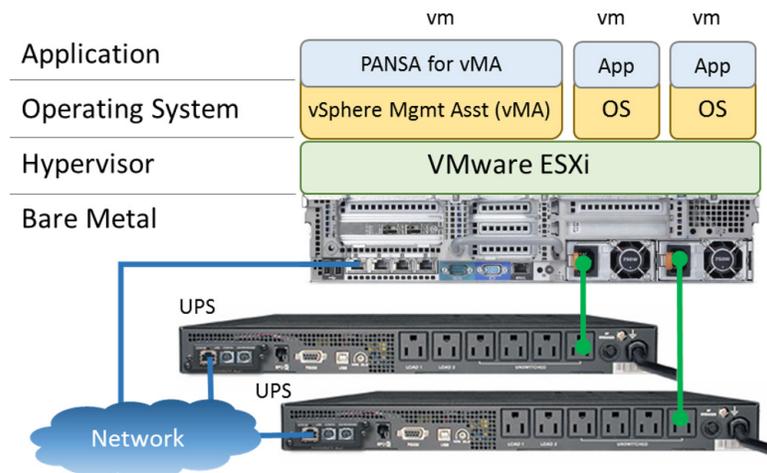
APPENDIX A – PANSA for VMware ESXi 5.x and 6.x

A.1.1 Installation

Step 1. Install VMware vSphere Management Assistant (vMA)

Visit **http://www.vmware.com/support/developer/vima/** for more information on downloading and installing the vMA on ESXi host machines.

Step 2. Install VMware Tools

VMware Tools must be installed on each virtual machine for graceful shutdowns whenever the host receives a power outage notification.

1. Open a vSphere Client and connect to an ESXi host server.

2. Click the "Virtual Machines" tab.

3. For each virtual machine:

    a. Right-click on the VM, select Power → Power On.

    b. Right-click on the VM, select Open Console.

    c. Right-click on the VM, select Guest → Install VMware Tools.

    d. Complete the VMware Tools installation via the virtual machine's console.

Note: The steps above apply to the VMware vSphere Client. Other clients (e.g. the vSphere Web Client) will have slightly different steps.

Step 3. Configure ESXi Shutdown

The ESXi host must be configured to allow automatic startup and shutdown of virtual machines (guest operating systems). The following steps provide information on configuring the ESXi host to gracefully shutdown the virtual machines.

Note: These steps are required for ESXi hosts that are NOT in a High-Availability cluster. For hosts in a High-Availability cluster, proceed to step 4.

1. Click the "Configuration" tab.

2. Select "Virtual Machine Startup/Shutdown".

3. Click "Properties", located in the panel's top right-hand corner.

4. Select the checkbox to allow virtual machines to start and stop automatically with the system.

5. Enter the appropriate default startup delay and default shutdown delay desired.

6. Choose "Guest Shutdown for Shutdown Action".

7. For each virtual machine that is to be shut down during a power outage:

    a. Select and highlight the VM.

    b. Click the "Move Up" button until the VM is listed in the Automatic Startup section.

8. Use the "Move Up" and "Move Down" buttons to order all VMs accordingly.

9. The following screen depicts a correctly configured Virtual Machine Startup and Shutdown dialog:
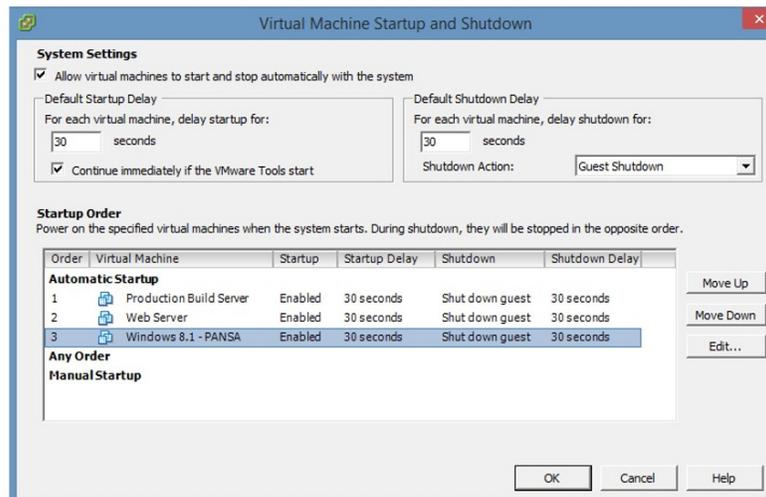
Note: The steps above apply to the VMware vSphere Client. Other clients (e.g. the vSphere Web Client) will have slightly different steps.

Step 4. Configure Firewall, WEBCARDLX and/or SNMPWEBCARD

Note: PANSA requires pre-installation configuration of firewall software settings, WEBCARDLX and/or SNMPWEBCARD settings.

• Open UDP ports 161, 162, 3664 and 3665 on your virtual machine's firewall software.

• The host computer for the device PANSA will be monitoring must also have UDP ports 161, 162, 3664 and 3665 open on its firewall software.

• If PANSA will be configured to monitor WEBCARDLX or SNMPWEBCARD, enable the card's SNMP access and define a read/write community string (the default community string for firmware version 12.04.0048 and above is **tripplite**).

Warning: Do not install more than one edition of a PowerAlert product on a single VM.

If a previous version of PANSA has been installed, Tripp Lite recommends you manually uninstall it prior to installing the new version.

Step 5. Install PANSA for vMA

The PowerAlert Network Shutdown Agent for vMA download contains all files required for installation and configuration onto the OpenSUSE-based vMA virtual machine.

1. Go to **tripplite.com/products/power-alert** to download the latest version of PANSA for vMA.

2. Upload the RPM package to the vMA VM using an SCP client (e.g. WinSCP).

3. Open a terminal on the vMA.

4. Change to the directory where the file is located and enter the following command: **sudo rpm -ivh <rpm-file-name> --nodeps**. Use the RPM man page to help you through any installation issues.

Note: Four VMware "script missing" warnings will appear after the installation completes. Please disregard these warnings.

PANSA will install to the directory **/var/tripplite/poweralert** and will place or install the daemon process for each software version within **/var/tripplite/poweralert/engine** .

A.1.2 Configuration

VMware's vMA virtual machine only offers a command line interface. Therefore, it is necessary to configure PANSA using its "paconfig. ini" textual configuration file. The instructions and details for modifying the configuration file are the same as described for the PANSA for Linux installation. Refer to **Section 3.3.1** for instructions on editing the configuration file.

PANSA is typically configured for shutting down an operating system; however, this may not be the case when running the configuration as a virtual machine. With virtual machine configurations, users often will opt to shut down the host ESXi server instead of the VM. Executing a script is usually required to perform this action. As such, the following [RemoteEvent:On Battery] section should be used for a virtual machine configuration instead of the instructions detailed in **Section 3.3.1**:

```
[RemoteEvent:On Battery]
EnableShutdown=false
ShutdownDelay=15
EnableExecuteCommand=true
ExecuteCommandDelay=5
ExecuteCommandOnSet=/home/vi-admin/shutdownesxi.sh
ExecuteCommandOnClear=
```

The difference in the configuration instructions in this section from the instructions in **Section 3.3.1** is the EnableShutdown value is set to "false" and the EnableExecuteCommand is set to "true". Doing this will cause PANSA to execute the specified OnSet script when the On Battery event is triggered, but not shut down the operating system. The shutdown configuration of the ESXi server described in **Step 3** will shut down the virtual machine on which PANSA is running. When performing this action, it is also important to configure the ESXi server to shut down this virtual machine last.

A.1.3 Modifying and Configuring Scripts

PANSA manages and monitors a device—typically a UPS—and then optionally executes a command or script that instructs the computer to perform some process upon receiving a critical event, such as a power outage. The PANSA installation provides sample command-line scripts that trigger the ESXi shutdown. However, these scripts must be modified and configured for use in your environment as follows:

1. From the vMA terminal, change to the /var/tripplite/poweralert directory **cd /var/tripplite/poweralert**

2. Copy the desired script to the /home/vi-admin directory **cp shutdownesxi.sh /home/vi-admin/shutdownesxi.sh** or **cp shutdownvms.sh /home/vi-admin/shutdownvms.sh**

   Note: Use shutdownesxi.sh if the ESXi servers are not part of an HA cluster and shutdownvms.sh if the ESXi servers are part of an HA cluster

3. Change to the home directory **cd /home/vi-admin**.

4. Change the script attributes by using the command **sudo chmod 755 shutdownesxi.sh** or **sudo chmod 755 shutdownvms.sh**

5. Edit the configuration file by typing **sudo vi shutdownesxi.sh** or **sudo vi shutdownvms.sh**

6. Press the Insert key to change vi to Insert mode and change the parameters in the selected script as required. See comments in the scripts or Appendix A in this document for details.

7. Save the changes to the script by typing **:wq<CR>**
   <CR>=carriage return (e.g. press the "Enter" or "Return" key on your keyboard).

Note: It is recommended that you test your modified script before use within PANSA to ensure successful performance.

Note: Each script can be extended to perform whatever actions are desired. Sample scripts, which include shutting down multiple hosts and shutting down hosts in High Availability (HA) clusters, can be found in Appendix A.

A.2 PANSA for Non-vMA Operating Systems

PANSA can be installed on any supported virtual or physical machine with network access to the ESXi hosts and provide the same functionality as the vMA. In addition to the prerequisites described in the vMA configuration, the VMware vSphere CLI will need to be installed on the same operating system as where PANSA will be installed. This section outlines the steps required to install PANSA on a non-vMA virtual machine.
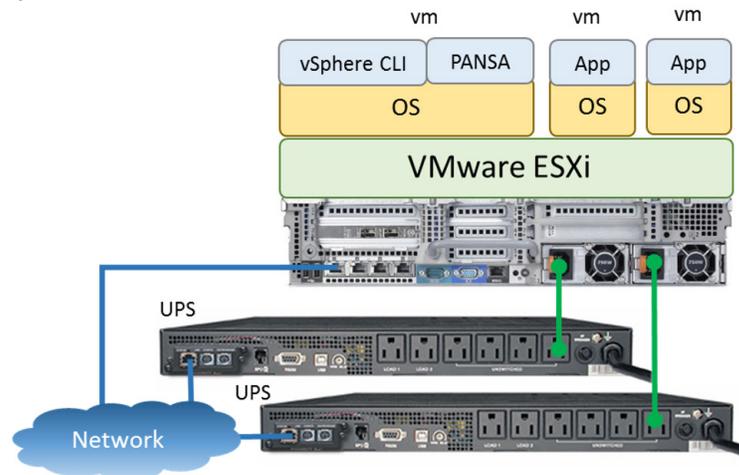
Step 1. Install VMware Tools on the target virtual machine (refer to Step 2 in **Section A.1.1** for details).

Step 2. Configure shutdown settings on the host ESXi server (refer to Step 3 in **Section A.1.1** for details).

Step 3. Install VMware's vSphere CLI.

VMware's vSphere CLI provides the same scripting functionality as is built into the vMA. Visit VMware's web site to download and install the vSphere CLI onto the target virtual machine.

Step 4. Install the version of PANSA respective to the target VM's operating system.

Note: Beginning with version 6.5 of the VMware vSphere CLI, users are required to install Perl (ActivePerl or Strawberry Perl version 5.14 or later as of this writing) separately from the VMware vSphere CLI. Users may prefer to install an earlier version of the VMware vSphere CLI instead. Currently, version 6.0, which includes Perl, can be found at the link below:

https://my.vmware.com/web/vmware/details?downloadGroup=VCLI600&productId=491

The PANSA download contains all files required for installation and configuration onto the VM or physical machine. If the VM or physical machine is using Windows, refer to **Section 2**. for installation and configuration instructions. In Windows, the sample script files will install with PANSA and be found at "C:\Program Files (x86)\TrippLite\PowerAlert". In Linux, the script files will install with PANSA and be found at /var/tripplite/poweralert. If the VM or physical machine is using Linux, refer to **Section 3**. for installation and configuration instructions.

A.3 PowerAlert Local and PowerAlert Office

This solution should only be used when a WEBCARDLX or SNMPWEBCARD is not available. PowerAlert Local (PAL) and PowerAlert Office (PAO) communicate directly with a UPS via an RS-232 serial or USB connection to determine when a critical shutdown event such as a power outage occurs. It can shut down the computer and execute a script, which instructs an ESXi host to shut down just like PANSA. The pre-requisites for this implementation are identical to those identified in **Section A.2**.
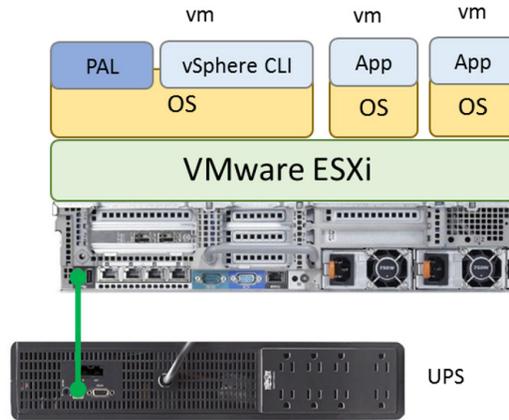
Figure A.3.1

Instructions on installing and configuring PAL and PAO are beyond the scope of this manual. For more information, refer to Tripp Lite's web site.

The same PANSA scripts work for PowerAlert Local, regardless if used for Windows or Linux. However, the same modifications to configure them for your environment must still be performed. Refer to **Section A.1.3** for more information on modifying and configuring scripts.

APPENDIX A – PANSA for VMware ESXi 5.x and 6.x

The PANSA installation provides several sample command-line scripts for both Windows and Linux operating systems. In Windows, the sample script files will be found at "C:\Program Files (x86)\TrippLite\PowerAlert". In Linux, the script files will be found at /var/tripplite/poweralert. This appendix describes these sample scripts. Also, refer to comments inside the file for further instructions on configuring the scripts for your environment.

Note: Each sample script shown in this appendix requires that user names and passwords be entered in their appropriate command line spaces. Normal alphanumeric characters can generally be written as is; however, some special characters such as !, #, % and $ need to be prefixed with a backslash ("\"). For example, the password "h@ppy" must be typed as "h\@ppy" in the command line for the script to function correctly.

A.4 Shutdown ESXi Server(s)

This script is used to shut down one or more ESXi servers, relying upon the pre-configured shutdown settings on each server to gracefully shut down the virtual machines running on each. This is the recommended script to use when installing PANSA for VMware in its recommended configuration (installing PANSA on a vMA).

In these scripts, the only edits required are updating the values after --server, --username, and --password. The --server value (10.0.025 in the example) should be the IP address of the ESXi host, the --username value (xxxx in the example) should be the username of the ESXi host, and the --password value (xxxx in the example) should be the password for the ESXi host. Uncomment additional lines as needed to shut down other ESXi hosts.

Linux Script: shutdownesxi.sh

```
#!/bin/bash
# Change directory to directory containing vicfg script files.
cd /usr/bin

# Resolves "server version unavailable" error caused by perl
# module wanting to check a certificate, which many not be
# available.
export PERL_LWP_SSL_VERIFY_HOSTNAME=0

# Trigger the host to shut down based upon its preconfigured
# settings. Be sure to change the server, username and password
# parameters before use.
./vicfg-hostops --server 10.0.0.25 --username xxxx --password xxxx --operation shutdown --force

# Duplicate the above line as needed to shut down additional
# ESXi servers.
#./vicfg-hostops --server 10.0.0.26 --username xxxx --password xxxx --operation shutdown --force
#./vicfg-hostops --server 10.0.0.27 --username xxxx --password xxxx --operation shutdown --force
#./vicfg-hostops --server 10.0.0.28 --username xxxx --password xxxx --operation shutdown --force
```

Windows Script: shutdownesxi.bat

```
#@echo off
REM Use this file in place of the shutdownesxi.bat file
REM included with PANSA 12.04.0058
REM Change directory to directory containing vicfg script files.
cd "C:\Program Files (x86)\VMware\VMware vSphere CLI\bin"
REM Users of 32 bit OS, uncomment the line below, comment out the line above
REM cd "c:\program files \vmware\vmware vsphere cli\bin"

REM Trigger the host to shut down based upon its preconfigured
REM settings. Be sure to change the server, username and password
REM parameters before use.
REM User Input required change server, username, and password values
perl vicfg-hostops.pl --server 10.0.0.25 --username xxxxx --password xxxxx --operation shutdown --force

REM Duplicate the above line as needed to shut down additional
REM ESXi servers.
REM perl vicfg-hostops.pl --server 10.0.0.26 --username xxxxx --password xxxxx --operation shutdown --force
REM perl vicfg-hostops.pl --server 10.0.0.27 --username xxxxx --password xxxxx --operation shutdown --force
REM perl vicfg-hostops.pl --server 10.0.0.28 --username xxxxx --password xxxxx --operation shutdown --force
```

A.5 Shutdown VMs on ESXi Server(s)

This script is used to forcibly shut down the virtual machines on one or more ESXi servers, and then shut down the ESXi server(s). This method does not rely upon the pre-configured shutdown settings on each server to shut down the virtual machines. This script is typically used when performing a shutdown for virtual machines in a high availability (HA) cluster.

Linux Script: shutdownvms.sh

```
#!/bin/sh
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/vmware/vma/lib64:/opt/vmware/vma/lib
export LD_LIBRARY_PATH
export PERL_LWP_SSL_VERIFY_HOSTNAME=0
SAVEIFS=$IFS
IFS=$(echo -en "\n\b")
# Use this script to shutdown ESXi hosts in an HA Cluster
# This script requires that the ESXi hosts are controllable by the vMA.
# You can check if the servers are added to the vMA by running
# the command "vifp listservers".
# You can add the ESXi servers by running the command
# "vifp addserver <host IP>" from the vMA terminal
# Designate the ESXi host(s) to shut down.
# User input required- Enter IP addresses for all ESXi hosts in the cluster.
hosts=(10.0.0.25 10.0.0.26 10.0.0.27 10.0.0.28)
# User input required- Input the exact virtual machine name for the vMA
# use \ to escape special characters in the name, e.g. \(
ups_vm="vSphere Management Assistant \(vMA\)"
# Change directory to directory containing vicfg script files
cd /usr/bin
for host in ${hosts[@]}; do
echo $host
source /opt/vmware/vma/bin/vifptarget -s $host
# Be sure to change the username and password parameters before use
# User input required- input the username and password for the hosts
for i in `vmware-cmd -l --username xxxx --password xxxx`; do
if [ `vmware-cmd $i getstate | egrep -c "on"` -eq 1 ]; then
echo $i
if [ `echo $i | egrep -c $ups_vm` -eq 1 ]; then
echo "Skip shut down of VMA"
else
echo "Shutting down $i"
vmware-cmd "$i" stop soft
fi
fi
done
source /opt/vmware/vma/bin/vifptarget -c
# User input required- input the username and password for the hosts
# Do NOT change the $host text
# Trigger the host server to shut down. Be sure to change the username
# and password parameters before use.
./vicfg-hostops --server $host --username xxxxx --password xxxxx --operation shutdown --force
done
```

Windows Script: shutdownvms.pl

```perl
#!/usr/bin/perl -w
# Use this script to shutdown ESXi Hosts in an HA Cluster from a
# Windows machine outside of the cluster
# Prerequisites for this script to work include the vSphere CLI being installed
# and, if using version 6.5 or later of the vSphere CLI, a compatible version of Perl e.g.
# ActivePerl or Strawberry Perl version 5.14
# Place this script at C:\Program Files (x86)\VMware\VMware vSphere CLI\bin and
# call this script from PANSA or PAL
# By default, most hosts will come back in Maintenance mode when powered back on after using
# this script
# Shut down the VMs << Uses vmware-cmd.pl
# Place the hosts in Maintenance mode <<Uses vicfg-hostops.pl
# Shut down the hosts <<Uses vicfg-hostops.pl
# Shut down vCenter (if vCenter is a vm you will need to connect to each host separately and shut it down.)
# << This script connects to each host separately
#use strict;
use warnings;
use 5.14.4;
use VMware::VIRuntime;
use VMware::VILib;
use VMware::VIExt;
my $host;
my $state;
my $toolsstate;
my @vmxcmd;
my $vmxcmd;
my @vmxlist;
my $vmxlist;
my $statecmd;
my $stop;
#my $vmwareBIN = "C:\Program Files (x86)\VMware\VMware vSphere CLI\bin";
my $vmwareCMD = "vmware-cmd.pl";
my $vicfgHostops = "vicfg-hostops.pl";
# User input required- Credientals for ESXi Servers
# Be sure to change the username and password parameters before use
# use \ to escape special characters, e.g. \@
my $loginCMD = "-U xxxx -P xxxxxx";
# User input required- Credientals for ESXi Servers
# Be sure to change the username and password parameters before use
# use \ to escape special characters, e.g. \@
my $loginHostops = "--username xxxx --password xxxxxx";
# Designate the ESXi host(s) to shut down.
# User input required- Enter IP addresses for all ESXi hosts in the cluster.
my @hosts = ( "10.0.0.25","10.0.0.26","10.0.0.27","10.0.0.28");
foreach my $host (@hosts) {
#say $host;
@vmxlist=();
$vmxlist="";
$vmxcmd = "$vmwareCMD -H $host $loginCMD -l";
#say $vmxcmd;
$vmxlist = `$vmxcmd`.$vmxlist;
#say $vmxlist;
$vmxlist =~ s/^\n//;
#say $vmxlist;
@vmxlist = split /\n/, $vmxlist;
foreach $vmxlist (@vmxlist) {
#getState (on, off, suspended)
$statecmd = "$vmwareCMD -H $host $loginCMD \"$vmxlist\" getstate";
#say $statecmd;
$state = `$statecmd`;
#say $state;
if($state =~ /(on)$/) {
#getToolsState
$toolsstate = `"$vmwareCMD" -H $host $loginCMD \"$vmxlist\" gettoolslastactive`;
#say $toolsstate;
if($toolsstate =~ /(1)$/) {
#shutdown vm gracefully
$stop = `"$vmwareCMD" -H $host $loginCMD \"$vmxlist\" stop soft`;
}
else {
```

```
#stop vm
`"$vmwareCMD" -H $host $loginCMD \"$vmxlist\" stop hard`;
}
}
}
}
# User input optional- change the wait period if appropriate.
# Value is in seconds
# insert wait period to complete vm shutdown
sleep 120;
#Place the host(s) in Maintenance mode
foreach my $host (@hosts) {
say $host;
$vmxcmd = `vicfg-hostops.pl --server $host $loginHostops -operation enter --force';
#say $vmxcmd;
}
# User input optional- change the wait period if appropriate.
# Value is in seconds
# insert wait period to complete maintenance mode
sleep 30;
#Shutdown the host(s)
foreach my $host (@hosts) {
say $host;
$vmxcmd = `vicfg-hostops.pl --server $host $loginHostops -operation shutdown --force';
#say $vmxcmd;
}
exit
```

Below is the content of the poweralert.conf file:

```
# ----------------------------------------------------------------------------
# Sample configuration file for Tripp Lite PowerAlert Applications:
#
# PAL   (PowerAlert Local)
# PANMS (PowerAlert Network Management System)
# PANSA (PowerAlert Network Shutdown Agent)
# ----------------------------------------------------------------------------

# Set to 'true' to daemonize (the default), 'false' to run as console
#
daemonize = true

# syslog configuration. PowerAlert will always log the initial setup and
# configuration results to the local syslog. By setting the following
# option to a value other than NONE, syslog will continue to run with
# the requested minimal log level
#
syslog.level = ERROR

# Log file configuration. PowerAlert can optionally log to the specified
# file and minimal log level
#
filelog.file = /var/tripplite/poweralert/log/poweralert.log
filelog.level = INFO

# When running in console mode, the default is to have PowerAlert log to
# standard out. This option allows you to set the minimum log level
#
consolelog.level = INFO

# When a system shutdown is requested by PowerAlert in response to one or
# more events, request this shutdown option.
#
# Valid values are:
#
#     HALT   - Shutdown and halt the system (shutdown -h)
#
#     POWEROFF - Shutdown and attempt to power down
#
#     BOOTMONITOR - Halt or drop into boot monitor on supported systems
#
#     NO_SHUTDOWN - Only send the warning messages; PowerAlert will still
#                   behave as if shut down and will need to be restarted.
#
shutdown.option = HALT

# Once a system shutdown has been invoked, this is the delay during which
# a root user can issue a "shutdown -c" to cancel the shutdown before it
# is processed by the operating system. This will not stop the PowerAlert
# shutdown - just the operating system shutdown. PowerAlert will still need
# to be restarted.
#
# The argument can have different formats. First, it can be an absolute
# time in the format hh:mm, in which hh is the hour (1 or 2 digits) and mm
# is the minute of the hour (in two digits). Second, it can be in the
# format +m, in which m is the number of minutes to wait. The word now is
# an alias for +0.
#
shutdown.time = now

# Message to send to all users when system shutdown activates
#
shutdown.message = PowerAlert starting system shutdown

# If set to a value greater than zero, init will be told to wait that
# number of seconds between sending processes the warning and the kill
# signal before changing to another run-level.
#
shutdown.initwait = 0
```

```
# When a system reboot is requested by PowerAlert in response to one or
# more events, request this reboot option.
#
# Valid values are:
#
#     REBOOT - Shutdown and reboot the system (shutdown -r)
#
#     FORCEFSCK - Shutdown and reboot the system, forcing fsck
#
#     SKIPFSCK - Shutdown and reboot the system, skipping fsck
#
#     NO_SHUTDOWN - Only send the warning messages; PowerAlert will still
#               behave as if shut down and will need to be restarted.
#
reboot.option = REBOOT

# Once a system reboot has been invoked, this is the delay during which
# a root user can issue a "shutdown -c" to cancel the shutdown before it
# is processed by the operating system.
#
# The argument can have different formats. First, it can be an absolute
# time in the format hh:mm, in which hh is the hour (1 or 2 digits) and mm
# is the minute of the hour (in two digits). Second, it can be in the
# format +m, in which m is the number of minutes to wait. The word now is
# an alias for +0.
#
reboot.time = now

# Message to send to all users when system reboot activates
#
reboot.message = PowerAlert rebooting system

# If set to a value greater than zero, init will be told to wait that
# number of seconds between sending processes the warning and the kill
# signal before changing to another run-level.
#
reboot.initwait = 0
```

How to Use testconnection.ps1

```
PS C:\Users\User> cd "C:\Program Files (x86)\TrippLite\PowerAlert\VMware_scripts\6.7andnewer\HA_cluster"
PS C:\Program Files (x86)\TrippLite\PowerAlert\VMware_scripts\6.7andnewer\HA_cluster> .\testconnection.ps1

Scope      ProxyPolicy    DefaultVIServerMode InvalidCertificateAction DisplayDeprecationWarnings WebOperationTimeout
                                                                                                  Seconds
-----      -----------    ------------------- ------------------------ -------------------------- -------------------
Session    UseSystemProxy Multiple            Ignore                   False                      300
User                                          Ignore                   False
AllUsers
Session    UseSystemProxy Multiple            Ignore                   False                      300
User                                          Ignore                   False
AllUsers
================================================================
Testing connection to vCenter: 10.0.0.34
================================================================
================================================================
Connection to vCenter 10.0.0.34 is successful
================================================================

Name    : 10.0.0.34
Version : 7.0.1
Build   : 17491160



PS C:\Program Files (x86)\TrippLite\PowerAlert\VMware_scripts\6.7andnewer\HA_cluster>
```

Both the HA_cluster and Not_HA script directories include a testconnection.ps1 PowerShell script. Use the script to test access.conf settings and connectivity between the computer running PANSA and vCenter (for the HA_cluster script) or ESXi servers (the NOT_HA script).

To run the test connection script, ensure it is in the same directory as a valid access.conf file, then perform the following steps:

1. Launch PowerShell

2. Change directory (cd) to the directory with the test connection script

3. Execute the script with the command

    .\testconnection.ps1

If the connection is successful, the output will include a message similar to:

Connection to vCenter <ADDRESS> is successful (for HA_cluster)
Or
Connection to <ADDRESS> ESXi host is successful (for NOT_HA)

Example access.conf for 1 ESXi Host

The configuration below is valid for an ESXi host at IP address 10.0.0.1 with a username of root and password of password.

hostslist=10.0.0.1
username=root
password=password
vmtimeout=240
vctimeout=120
vcentername=
upsvmname=

Example access.conf for 3 ESXi Hosts

The configuration below is valid for 3 ESXi hosts at IP addresses of 10.0.0.1, 10.0.0.2, and 10.0.0.3 with a username of root and password of password123. In this scenario, PANSA should run outside of the hosts or on the last ESXi host in the list (10.0.0.3).

hostslist=10.0.0.1,10.0.0.2,10.0.0.3
username=root
password=password123
vmtimeout=240
vctimeout=120
vcentername=
upsvmname=

APPENDIX C – VMware 6.7-7.x Sample Scripts and Files

Example access.conf for 3 ESXi Hosts in a Cluster, vCenter in Cluster

The access.conf configuration below is valid for an HA cluster with:

• A vCenter virtual machine with an IP of 10.0.0.4 that runs in the cluster and has a vcenterusername of administrator@vsphere.local and a vcenterpassword of vcenterpass123

• 3 ESXi hosts at IP addresses of 10.0.0.1, 10.0.0.2, and 10.0.0.3 with a username of root and password of password123

• A 4-minute timeout (vmtimeout) before forcibly shutting down any unresponsive virtual machines other than vCenter

• A 2-minute timeout (vctimeout) before forcibly shutting down the vCenter virtual machine if it is unresponsive

```
vcenter=10.0.0.4
vcentername=VMware vCenter Server
vcentershutdown=true
vcenterusername=administrator@vsphere.local
vcenterpassword=vcenterpass123
hostslist=10.0.0.1,10.0.0.2,10.0.0.3
username=root
password=password
vmtimeout=240
vctimeout=120
upsvmname=
```

Example access.conf for 3 ESXi Hosts in a Cluster, vCenter Outside of Cluster

If vCenter is running on a hypervisor outside of the cluster, set the vcentershutdown value to false. For example, the access.conf configuration below is valid for an HA cluster with:

• A vCenter virtual machine with an IP of 10.0.0.4 that runs outside of the cluster and has a vcenterusername of administrator@vsphere.local and a vcenterpassword of vcenterpass123

• 3 ESXi hosts at IP addresses of 10.0.0.1, 10.0.0.2, and 10.0.0.3 with a username of root and password of password123

• A 4-minute timeout (vmtimeout) before forcibly shutting down any unresponsive virtual machines other than vCenter

• A 2-minute timeout (vctimeout) before forcibly shutting down the vCenter virtual machine if it is unresponsive

```
vcenter=10.0.0.4
vcentername=VMware vCenter Server
vcentershutdown=true
vcenterusername=administrator@vsphere.local
vcenterpassword=vcenterpass123
hostslist=10.0.0.1,10.0.0.2,10.0.0.3
username=root
password=password
vmtimeout=240
vctimeout=120
upsvmname=
```

Standalone (Not in an HA Cluster) Default Scripts and access.conf Files

access.conf

```
# BASIC CONFIGURATION #########################
# All users should modify hostslist, username, and password.
# For the hostlist value input the IP addresses or FQDNs of all ESXi hosts to be shut down.
# Addresses should be separated by a comma, no space.
# If PowerAlert is on an ESXi host that will be shut down, place the address of the ESXi host running PowerAlert last.
hostslist=10.0.0.31,10.0.0.32,10.0.0.33
# Input the username and password of a user that can perform shutdown of Virtual Machines and ESXi hosts.
username=root
password=password
# END BASIC CONFIGURATION #####################
# ADVANCED ####################################
# CONFIGURATION ###############################
# BELOW #######################################
# ############################################
# The vmtimeout is not used in standard non-HA configurations, for advanced configuration only.
# Input the time in seconds to wait for graceful shutdown of Virtual Machines
# Default is 240 seconds
vmtimeout=240
# The vctimeout is not used in standard non-HA configurations, for advanced configuration only.
# Input the time in seconds to wait for graceful shutdown of vCenter VM
# Default is 120 seconds
vctimeout=120
```

Leave vcentername empty for standard non-HA cluster configurations.
For select advanced configurations, input Display Name of vCenter server VM (ususally VMware vCenter Server).
vcentername=
Leave upsvmname empty for standard non-HA cluster configurations.
For select advanced configurations, input the Display Name of the VM running PowerAlert software.
upsvmname=

shutdownesxi.bat

echo . | PowerShell.exe -ExecutionPolicy ByPass -command "./shutdownesxi.ps1" *> .\shutdownesxi.log

shutdownesxi.ps1

```
#####################################################################################
# Shutdown ESXi host(s)
#
# This script will loop through a list of ESXi hosts and initiate shutdown
# commands to the vm's residing on them. If VMware Tools is installed, the
# script will attempt to do a graceful shutdown. If VMware tools is not
# installed a hard power off will be performed. If PANSA is installed as VM
# on ESXi host it must be on the last ESXi host to shut down and remain in
# running state until the end of the process. PANSA VM name must in the $upsvmname
# variable and also must be on the last host listed in the $hosts variable.
#
#####################################################################################

# Import variables
$ScriptDir = Split-Path $script:MyInvocation.MyCommand.Path
$ScriptDir += "\access.conf"
Get-Content -Path $ScriptDir | Foreach-Object{
   $var = $_.Split('=')
   New-Variable -Name $var[0] -Value $var[1]
}

$Inputstring = $hostslist
$hosts = $InputString.Split(",")

# Ignore certificate warning
Set-PowerCLIConfiguration -InvalidCertificateAction Ignore -Confirm: $false
Set-PowerCLIConfiguration -DisplayDeprecationWarnings $false -Confirm:$false

# Loop through each host
Foreach ($esxhost in $hosts)
{

Write-Host "======================================================================"
Write-Host "Testing connection to ESXi host:" $esxhost
Write-Host "======================================================================"

        Connect-VIServer $esxhost -username $username -password $password -force -WarningAction SilentlyContinue -ErrorAction
SilentlyContinue
        if($?) {
        Write-Host "======================================================================"
        Write-Host "Connection to $esxhost ESXi host is successful" -Foregroundcolor Green
        Write-Host "======================================================================"
        }
        else {
        Write-Host "======================================================================"
        Write-Host "Connection to $esxhost ESXi host failed" -Foregroundcolor Red
        Write-Host "======================================================================"
        }
```

```
# Shut down the vCenter
$vcentershutdownvm = (Get-VM | where { $_.PowerState -eq "PoweredOn" -and $_.Name -eq "$vcentername"})
          if ($vcentershutdownvm -match "$vcentername")
          {
          Write-Host "======================================================================="
          Write-Host "Shut down the vCenter on the ESXi host" -Foregroundcolor Green
          Write-Host "======================================================================="
          Get-VM "$vcentername" | Shutdown-VMGuest -Confirm:$false
          # Amount of time to wait before assuming the remaining powered on guests will be shut down forcebly
          $waittime = $vctimeout #Seconds
          $time = (Get-Date).TimeofDay
          do {
     # Wait for the VMs to be shutdown cleanly
     Start-Sleep -s 1.0
     $timeleft = $waittime - ($newtime.TotalSeconds)
     $numvms = (Get-VM | Where {$_.PowerState -eq "PoweredOn" -and $_.Name -eq "$vcentername"}).Count
     Write-Host "Waiting for shutdown of $numvms VMs or until $timeleft seconds"
     $newtime = (Get-Date).TimeofDay - $Time
     } until ((@(Get-VM | Where {$_.PowerState -eq "PoweredOn" -and $_.Name -eq "$vcentername"}).Count) -eq 0 -or ($newtime).TotalSeconds -ge
$waittime)
          }
          else
          {
          Write-Host "======================================================================="
          Write-Host "There is no vCenter on this host" -Foregroundcolor Yellow
          Write-Host "======================================================================="
          }

# Uncomment lines 81-146 if you want to use script actions for VMs shutdown instead of "VM Startup/Shutdown Settings" on host(s).

$currentesxhost = get-vmhost $esxhost

#if($currentesxhost) {
#          Write-Host "======================================================================="
#          Write-Host "Proccesing ESXi host $currentesxhost" -Foregroundcolor Green
#          Write-Host "======================================================================="
#          }
#          else
#          {
#          Write-Host "======================================================================="
#          Write-Host "Failed to find ESXi host $currentesxhost" -Foregroundcolor Red
#          Write-Host "======================================================================="
#          exit $lastexitcode
#          }

## Loop through each vm on host
#          Foreach ($vm in ($currentesxhost | Get-VM | where { $_.PowerState -eq "PoweredOn" -and $_.Name -notlike "*vCLS*"}))
#          {
#          Write-Host "======================================================================="
#          Write-Host "Processing $vm" -Foregroundcolor Green
#          Write-Host "======================================================================="

## If this is a VM where UPS software is running, it must be shut down last
#if ($upsvmname -contains $vm)
#          {
#          Write-Host "======================================================================="
#          Write-Host "$vm will be shut down last" -Foregroundcolor Yellow
#          Write-Host "======================================================================="
#          }
#          else
#          {
#          Write-Host "======================================================================="
#          Write-Host "Checking VMware Tools...." -Foregroundcolor Green
#          Write-Host "======================================================================="
#          $vminfo = get-view -Id $VM.ID
```

```
## If VMware tools is not installed
#          if ($vminfo.config.Tools.ToolsVersion -eq 0)
#          {
#          Write-Host "======================================================================="
#          Write-Host "$vm doesn't have vmware tools installed, it will be shut down forcibly" -Foregroundcolor Yellow
#          Write-Host "======================================================================="

## Hard Power Off
#          Stop-VM $vm -confirm:$false -ErrorAction Stop -RunAsync
#          }
#          else
#          {
#          Write-Host "======================================================================="
#          Write-Host "$vm has vmware tools installed, it will be shut down gracefully" -Foregroundcolor Green
#          Write-Host "======================================================================="

## Power off gracefully
#$vmshutdown = $vm | shutdown-VMGuest -Confirm:$false -ErrorAction Stop
#          }
#          }
#          }
## Amount of time to wait before assuming the remaining powered on guests will be shut down forcibly
#$waittime = $vmtimeout #Seconds
#$time = (Get-Date).TimeofDay
#do {
#    # Wait for the VMs to be shutdown cleanly
#    Start-Sleep -s 1.0
#    $timeleft = $waittime - ($newtime.TotalSeconds)
#    $numvms = (Get-VM | Where {$_.PowerState -eq "PoweredOn" -and $_.Name -notlike "*vCLS*" -and $_.Name -ne "$upsvmname" -and $_.Guest.
Extensiondata.GuestState -ne "notRunning"}).Count
#    Write-Host "Waiting for shutdown of $numvms VMs or until $timeleft seconds"
#    $newtime = (Get-Date).TimeofDay - $time
#    } until ((@(Get-VM | Where {$_.PowerState -eq "PoweredOn" -and $_.Name -notlike "*vCLS*" -and $_.Name -ne "$upsvmname" -and $_.Guest.
Extensiondata.GuestState -ne "notRunning"}).Count) -eq 0 -or ($newtime).TotalSeconds -ge $waittime)

# Shut down the ESXi host

          Write-Host "======================================================================="
    Write-Host "Initiating host shutdown" -Foregroundcolor Green
          Write-Host "======================================================================="

          Write-Host "======================================================================="
          Write-Host "Shutting down the ESXi host" -Foregroundcolor Green
          Write-Host "======================================================================="
          Stop-VMHost $currentesxhost -Force -Confirm: $false -ErrorAction Stop -RunAsync
          while((Get-VMHost | where { $_.PowerState -ne "PoweredOn" -and $_.Name -eq $currentesxhost}).Count -ne 0)
                    {
                    Start-Sleep -s 1
                    }

          Write-Host "======================================================================="
          Write-Host "$currentesxhost ESXi host has shut down successfully" -Foregroundcolor Green
          Write-Host "======================================================================="
}
Write-Host "======================================================================="
Write-Host "Shutdown of all ESXi host(s) complete"
Write-Host "======================================================================="
```

testconnection.ps1 for Non-HA Environments

```
#####################################################################################
# Test connection to ESXi host(s)
#
# This script will loop through a list of ESXi hosts. Will be performed check as
# provided credentials in access.conf are valid and also provide ESXi version.
#
#####################################################################################

# Import variables
$ScriptDir = Split-Path $script:MyInvocation.MyCommand.Path
$ScriptDir += "\access.conf"
Get-Content -Path $ScriptDir | Foreach-Object{
   $var = $_.Split('=')
   New-Variable -Name $var[0] -Value $var[1]
}

$Inputstring = $hostslist
$hosts = $InputString.Split(",")

# Ignore certificate warning
Set-PowerCLIConfiguration -InvalidCertificateAction Ignore -Confirm: $false
Set-PowerCLIConfiguration -DisplayDeprecationWarnings $false -Confirm:$false

# Loop through each host
Foreach ($esxhost in $hosts)
{

Write-Host "======================================================================="
Write-Host "Testing connection to ESXi host:" $esxhost
Write-Host "======================================================================="

        Connect-VIServer $esxhost -Force -username $username -password $password
        if($?) {
        Write-Host "======================================================================="
        Write-Host "Connection to $esxhost ESXi host is successful" -Foregroundcolor Green
        Write-Host "======================================================================="
        }
        else {
        Write-Host "======================================================================="
        Write-Host "Connection to $esxhost ESXi host failed" -Foregroundcolor Red
        Write-Host "======================================================================="
        exit $lastexitcode
        }

        Get-VMHost | select Name,
                @{N='ESXi version';E={"$($_.Version) $($_.Build)"}}
}
```

HA Cluster Default Scripts and access.conf Files
access.conf

```
# Input the IP address or FQDN of the vCenter appliance.
vcenter=10.0.0.34
# Input the Display Name of vCenter server VM (ususally VMware vCenter Server).
vcentername=VMware vCenter Server
# The vcentershutdown option determines if the vCenter appliance will be shut down
# Valid values are: true or false
# If vCenter is running on an ESXi host which will be shut down, leave the value set to true
# If vCenter is running off the ESXi hosts that will be shut down, set the value to false
vcentershutdown=true
# Input the username and password of a vCenter user able to perform shutdown of Virtual Machines via vCenter.
vcenterusername=administrator@vsphere.local
vcenterpassword=password
# Input the IP addresses or FQDNs of all ESXi hosts. Addresses should be separated by a comma, no space.
# If PowerAlert is running on a host in the cluster (not typical for HA), place that host last in the list
hostslist=10.0.0.32,10.0.0.33
# Input the username and password of an ESXi user that can perform shutdown of Virtual Machines and ESXi hosts.
username=root
password=password
# Input the time in seconds to wait for graceful shutdown of Virtual Machines
```

40

```
# Default is 240 seconds
vmtimeout=240
# Input the time in seconds to wait for graceful shutdown of vCenter VM
# Default is 120 seconds
vctimeout=120
# ADVANCED CONFIGURATION
# Standard cases should leave upsvmname blank.
# For select advanced cases, provide the display Name of VM where PowerAlert software is running if it's in the cluster being shut down.
upsvmname=
```

shutdownvcenter.bat

```
echo . | PowerShell.exe -ExecutionPolicy ByPass -command "./shutdownvcenter.ps1" *> .\shutdownvcenter.log
```

shutdownvcenter.ps1

```
################################################################################
# Shutdown VMs and vCenter cluster(s)
#
# This script will loop through a list of cluster(s) and initiate shutdown
# commands to the vm's residing on them. If VMware Tools is installed, the
# script will attempt to do a graceful shutdown. If VMware tools is not
# installed a hard power off will be performed. If PANSA is installed as VM
# on ESXi host it must be on the last ESXi host to shut down and remain in
# running state until the end of the process. PANSA VM name must in the $upsvmname
# variable and also must be on the last host listed in the $hosts variable.
#
################################################################################

# Import variables
$ScriptDir = Split-Path $script:MyInvocation.MyCommand.Path
$ScriptDir += "\access.conf"
Get-Content -Path $ScriptDir | Foreach-Object{
    $var = $_.Split('=')
    New-Variable -Name $var[0] -Value $var[1]
}

$Inputstring = $hostslist
$hosts = $InputString.Split(",")

# Ignore certificate warning
Set-PowerCLIConfiguration -InvalidCertificateAction Ignore -Confirm: $false
Set-PowerCLIConfiguration -DisplayDeprecationWarnings $false -Confirm:$false
Set-PowerCLIConfiguration -ProxyPolicy NoProxy -Scope Session -Confirm:$false

# Testing connection to vCenter
Write-Host "===================================================================="
Write-Host "Testing connection to vCenter:" $vcenter
Write-Host "===================================================================="

        Connect-VIServer $vcenter -username $vcenterusername -password $vcenterpassword -force -WarningAction SilentlyContinue -ErrorAction
SilentlyContinue | Out-Null
                if($?) {
                Write-Host "===================================================================="
                Write-Host "Connection to vCenter $vcenter is successfull" -Foregroundcolor Green
                Write-Host "===================================================================="
                }
                else
                {
                Write-Host "===================================================================="
                Write-Host "Connection to vCenter $vcenter failed" -Foregroundcolor Red
                Write-Host "===================================================================="
                }
```

```
$clusters = Get-Cluster
ForEach ($cluster in $clusters)
{
if($cluster) {
           Write-Host "======================================================================="
           Write-Host "Proccesing cluster $cluster" -Foregroundcolor Green
           Write-Host "======================================================================="
           }
           else
           {
           Write-Host "======================================================================="
           Write-Host "Failed to find cluster $cluster" -Foregroundcolor Red
           Write-Host "======================================================================="
           }

#Retrieving a list of powered on guests
Write-Host "======================================================================="
Write-Host "Retrieving a list of powered on guests" -Foregroundcolor Green
Write-Host "======================================================================="
$poweredonguests = Get-VM -Location $cluster | where-object {$_.PowerState -eq "PoweredOn" -and $_.Name -notlike "*vCLS*" -and $_.Name -ne
$vcentername -and $_.Name -ne $upsvmname}

#Hard Power Off
ForEach ($guest in $poweredonguests)
{
    Write-Host "======================================================================="
           Write-Host "Processing $guest ...." -ForegroundColor Green
           Write-Host "======================================================================="
           Write-Host "======================================================================="
    Write-Host "Checking for VMware tools install" -Foregroundcolor Green
           Write-Host "======================================================================="
    $guestinfo = get-view -Id $guest.ID
    if ($guestinfo.config.Tools.ToolsVersion -eq 0)
    {
        Write-Host "======================================================================="
                     Write-Host "$guest doesn't have vmware tools installed, it will be shut down forcibly" -ForegroundColor Yellow
                     Write-Host "======================================================================="
        Stop-VM $guest -confirm:$false
    }
    else
    {
           Write-Host "======================================================================="
           Write-Host "$guest have vmware tools installed, it will be shut down gracefully" -Foregroundcolor Green
           Write-Host "======================================================================="
    $vmshutdown = $guest | shutdown-VMGuest -Confirm:$false
    }
}
# Amount of time to wait before assuming the remaining powered on guests will be shut down forcibly
                     $waittime = $vmtimeout #Seconds
                     $time = (Get-Date).TimeofDay
                              do {
                              # Wait for the VMs to be shutdown cleanly
                              Start-Sleep -s 1.0
                              $timeleft = $waittime - ($newtime.TotalSeconds)
                              $numvms = (Get-VM -Location $cluster | Where {$_.PowerState -eq "PoweredOn" -and $_.Name -notlike "*vCLS*" -and
$_.Name -ne $vcentername -and $_.Name -ne $upsvmname}).Count
                              Write-Host "Waiting for shutdown of $numvms VMs or until $timeleft seconds"
                              $newtime = (Get-Date).TimeofDay - $time
                              } until ((@(Get-VM -Location $cluster | Where {$_.PowerState -eq "PoweredOn" -and $_.Name -notlike "*vCLS*" -and
$_.Name -ne $vcentername -and $_.Name -ne $upsvmname}).Count) -eq 0 -or ($newtime).TotalSeconds -ge $waittime)
}

# Ignore certificate warning
Set-PowerCLIConfiguration -InvalidCertificateAction Ignore -Confirm: $false
Set-PowerCLIConfiguration -DisplayDeprecationWarnings $false -Confirm:$false
Set-PowerCLIConfiguration -ProxyPolicy NoProxy -Scope Session -Confirm:$false
```

```powershell
Foreach ($esxhost in $hosts)
{
Write-Host "======================================================================="
Write-Host "Testing connection to ESXi host:" $esxhost
Write-Host "======================================================================="

        Connect-VIServer $esxhost -username $username -password $password -force -WarningAction SilentlyContinue -ErrorAction
SilentlyContinue
        if($?) {
        Write-Host "======================================================================="
        Write-Host "Connection to $esxhost ESXi host is successful" -Foregroundcolor Green
        Write-Host "======================================================================="
        }
        else {
        Write-Host "======================================================================="
        Write-Host "Connection to $esxhost ESXi host failed" -Foregroundcolor Red
        Write-Host "======================================================================="
        }

# Shut down the vCenter
$vcentershutdownvm = (Get-VM | where { $_.PowerState -eq "PoweredOn" -and $_.Name -eq $vcentername})
        if (($vcentershutdownvm -match "$vcentername") -and ($vcentershutdown -match "true"))
        {
        Write-Host "======================================================================="
        Write-Host "Shut down the vCenter on the ESXi host" -Foregroundcolor Green
        Write-Host "======================================================================="
        Get-VM "$vcentername" | Shutdown-VMGuest -Confirm:$false
# Amount of time to wait before assuming the remaining powered on guests will be shut down forcibly
$waittime = $vctimeout #Seconds
$time = (Get-Date).TimeofDay
do {
    # Wait for the VMs to be shutdown cleanly
    Start-Sleep -s 1.0
    $timeleft = $waittime - ($newtime.TotalSeconds)
    $numvms = (Get-VM | Where {$_.PowerState -eq "PoweredOn" -and $_.Name -eq $vcentername}).Count
    Write-Host "Waiting for shutdown of $numvms VMs or until $timeleft seconds"
    $newtime = (Get-Date).TimeofDay - $Time
    } until ((@(Get-VM | Where {$_.PowerState -eq "PoweredOn" -and $_.Name -eq $vcentername }).Count) -eq 0 -or ($newtime).TotalSeconds -ge
$waittime)
        }
        else
        {
        Write-Host "======================================================================="
        Write-Host "There is no vCenter on this host or shutdown is disabled in access.conf" -Foregroundcolor Yellow
        Write-Host "======================================================================="
        }

    Write-Host "======================================================================="
    Write-Host "Initiating host shutdown" -Foregroundcolor Green
        Write-Host "======================================================================="

# Shut down the ESXi host

        $currentesxhost = get-vmhost $esxhost

        Write-Host "======================================================================="
        Write-Host "Shutting down the ESXi host" -Foregroundcolor Green
        Write-Host "======================================================================="
        Stop-VMHost $currentesxhost -Force -Confirm: $false -ErrorAction Stop -RunAsync
        while((Get-VMHost | where { $_.PowerState -ne "PoweredOn" -and $_.Name -eq $currentesxhost}).Count -ne 0)
                {
                Start-Sleep -s 1
                }

        Write-Host "======================================================================="
        Write-Host "$currentesxhost ESXi host has shut down successfully" -Foregroundcolor Green
        Write-Host "======================================================================="
}
Write-Host "======================================================================="
Write-Host "Shutdown of all ESXi host(s) complete"
Write-Host "======================================================================="
```

testconnection.ps1 for HA Environments

```
##############################################################################
# Test connection to vCenter
#
# This script will check connection to vCenter server. Will be performed check as
# provided credentials in access.conf are valid and also provide vCenter version.
#
##############################################################################

# Import variables
$ScriptDir = Split-Path $script:MyInvocation.MyCommand.Path
$ScriptDir += "\access.conf"
Get-Content -Path $ScriptDir | Foreach-Object{
   $var = $_.Split('=')
   New-Variable -Name $var[0] -Value $var[1]
}

$Inputstring = $hostslist
$hosts = $InputString.Split(",")

# Ignore certificate warning
Set-PowerCLIConfiguration -InvalidCertificateAction Ignore -Confirm: $false
Set-PowerCLIConfiguration -DisplayDeprecationWarnings $false -Confirm:$false

Write-Host "======================================================================="
Write-Host "Testing connection to vCenter:" $vcenter
Write-Host "======================================================================="

        Connect-VIServer $vcenter -username $vcenterusername -password $vcenterpassword -force -WarningAction SilentlyContinue -ErrorAction
SilentlyContinue | Out-Null
                if($?) {
                Write-Host "======================================================================="
                Write-Host "Connection to vCenter $vcenter is successful" -Foregroundcolor Green
                Write-Host "======================================================================="
                }
                else
                {
                Write-Host "======================================================================="
                Write-Host "Connection to vCenter $vcenter failed" -Foregroundcolor Red
                Write-Host "======================================================================="
                exit $lastexitcode
                }

                $Global:DefaultVIServers | select Name, Version, Build
```