APC PowerChute^{TM} shutdown solution for Dell EqualLogic Storage Array

By David Grehan, Sarah Jane Hannon

ABSTRACT

PowerChute^{TM} Network Shutdown software works in conjunction with the UPS Network Management Card to provide graceful network-based shutdown in the event of a power outage to prevent system damage and ensure data protection.

This Application Note contains step-by-step instructions with screenshots for customers to ensure the safe shutdown of their Dell EqualLogic Storage Array using APC PowerChute Network Shutdown.

Applications

IT Server Rooms, Data Centers, Remote Branch Offices, Distributed Networks.

Customer Benefits

- Graceful network-based shutdown
- Run command file capability
- Event logging
- Secure communications
- Browser accessible
- Redundant UPS configuration support
- Parallel UPS configuration support
- HTTPS communications
- IPv6 support

Introduction

If you have a Dell EqualLogic Storage Array in your set-up that you want to safely shut down in the event of a power outage, this paper provides a step-by-step guide on how to configure this using an APC UPS, Switched Rack PDU and PowerChute Network Shutdown software.

This paper is structured under the following headings:

- Minimum Requirements
- Basic Set-Up
- Required Scripts
- Configuration

Getting Started

Minimum Hardware Requirements

- Dell EqualLogic Storage Array
- APC UPS capable of supporting Server Load.
- APC Switched Rack PDU connected to the network with latest firmware*.
- APC UPS Network Management Card (to connect UPS to network and communicate UPS status to PowerChute software)

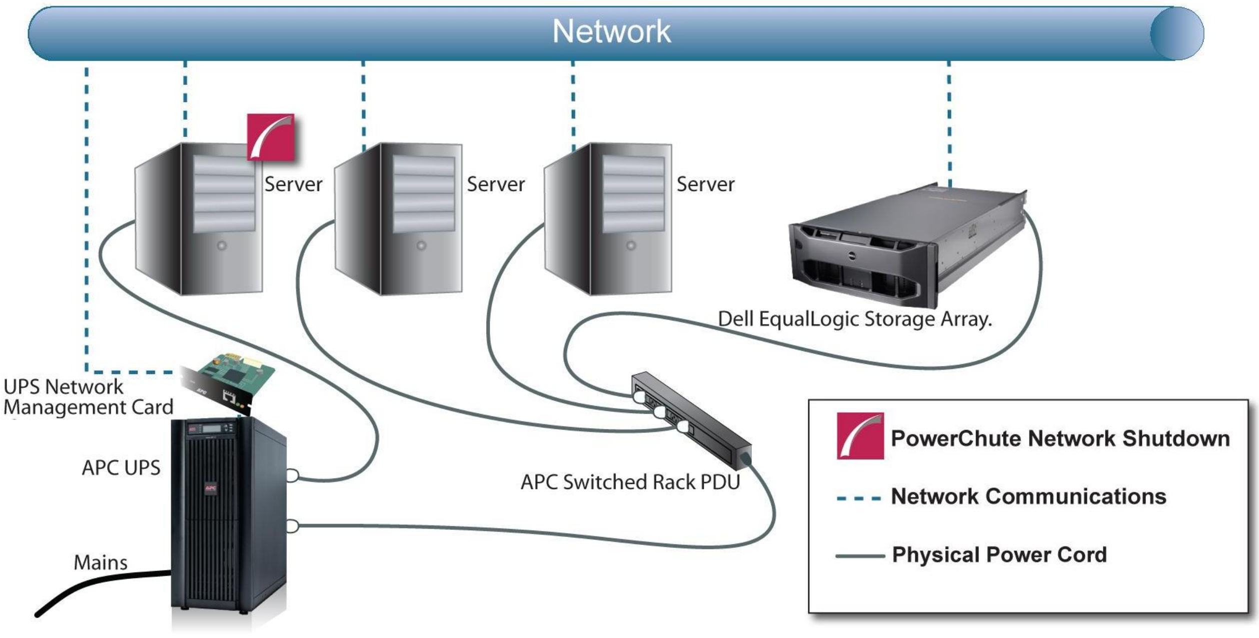
Minimum Software Requirements

- PowerChute Network Shutdown v4.1 - download from http://www.apc.com/tools/download/index.cfm .
- Strawberry Perl (32 or 64 bit version) from www.strawberryperl.com in order to run the two shutdown scripts.

* This scenario was tested with: AP7900 Switched Rack PDU with firmware rpduv3.7.4 (AOS 3.7.4).

Basic Set-Up

- One APC UPS powers the Dell EqualLogic Storage Array, the Switched Rack PDU and all servers.
- All servers are powered by the PDU except for a separate control server that will shut down the Storage Array and PDU.
- This control server is powered directly by the UPS.
- PowerChute Network Shutdown v4.1 is installed on all servers in the configuration.


Network
Server
Server
Server
Dell EqualLogic Storage Array.
UPS Network Management Card
APC UPS
Mains
APC Switched Rack PDU
PowerChute Network Shutdown
- - - Network Communications
—— Physical Power Cord

Required Scripts

Note: All scripts have been tested in a controlled environment by APC and Dell. These are sample scripts to be used at your own risk after first testing them in your own test environment before production use.

1) Script to shut down the Dell EqualLogic Storage Array

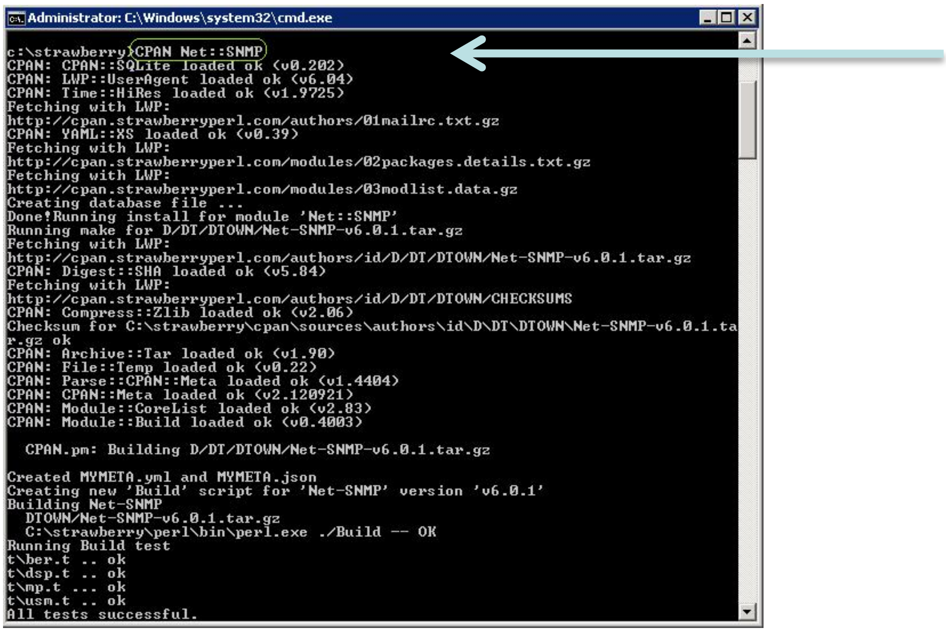
- Dell EqualLogic Scripting Tool are specific to the firmware version that's running on the Storage Array.
- Scripts can be downloaded from the EqualLogic Support site (www.EqualLogic.com/support). A log in is required.
- Host Scripting Tools for Windows & Linux can be found by selecting PS Series Firmware from the Downloads section in the top banner.
- Next, click on the Download Page for firmware version currently running on your array.

2) Script to turn off APC Switched Rack PDU

- Download the 32bit or 62bit strawberry Perl version from the web site free of charge (no log in required).
- We used a Net::SNMP module to issue SNMP GET/SET commands to the PDU, but other command line SNMP tools are available.
- The script sends an SNMP SET command to turn off the PDU outlets.

Steps for configuring PERL/editing the APC script:

1) After you have installed Strawberry PERL open a command prompt and type the following command:
CPAN Net::SNMP



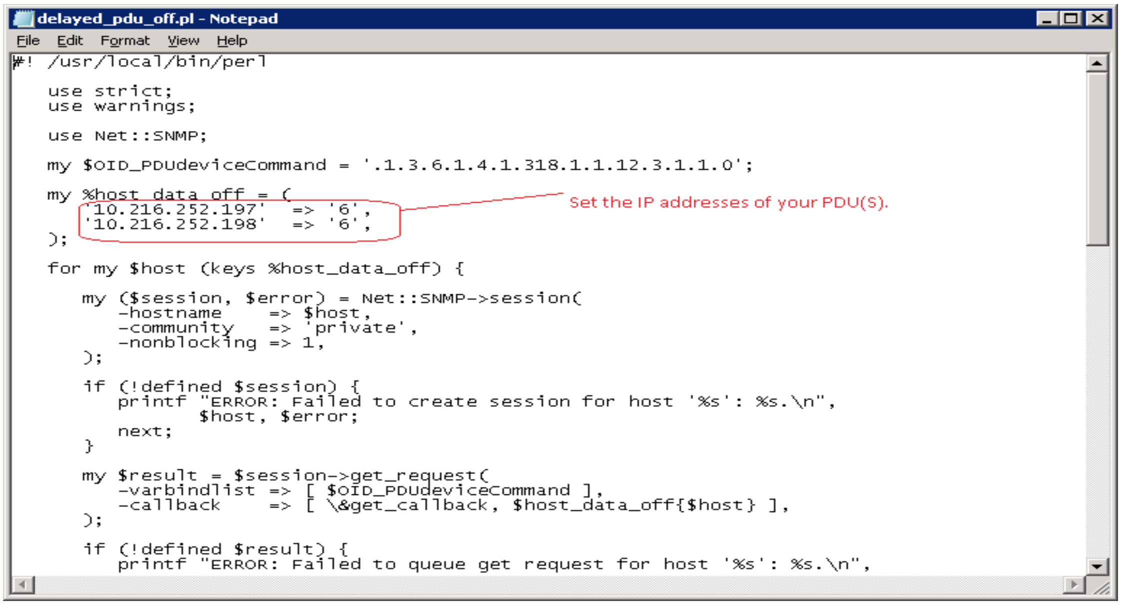
```
c:\strawberry>CPAN Net::SNMP
CPAN: CPAN::SQLite loaded ok (v0.202)
CPAN: LWP::UserAgent loaded ok (v6.04)
CPAN: Time::HiRes loaded ok (v1.9725)
Fetching with LWP:
http://cpan.strawberryperl.com/authors/01mailrc.txt.gz
CPAN: YAML::XS loaded ok (v0.39)
Fetching with LWP:
http://cpan.strawberryperl.com/modules/02packages.details.txt.gz
Fetching with LWP:
http://cpan.strawberryperl.com/modules/03modlist.data.gz
Creating database file ...
Done!Running install for module 'Net::SNMP'
Running make for D/DT/DTOWN/Net-SNMP-v6.0.1.tar.gz
Fetching with LWP:
http://cpan.strawberryperl.com/authors/id/D/DT/DTOWN/Net-SNMP-v6.0.1.tar.gz
CPAN: Digest::SHA loaded ok (v5.84)
Fetching with LWP:
http://cpan.strawberryperl.com/authors/id/D/DT/DTOWN/CHECKSUMS
CPAN: Compress::Zlib loaded ok (v2.06)
Checksum for C:\strawberry\cpan\sources\authors\id\D\DT\DTOWN\Net-SNMP-v6.0.1.ta
r.gz ok
CPAN: Archive::Tar loaded ok (v1.90)
CPAN: File::Temp loaded ok (v0.22)
CPAN: Parse::CPAN::Meta loaded ok (v1.4404)
CPAN: CPAN::Meta loaded ok (v2.120921)
CPAN: Module::CoreList loaded ok (v2.83)
CPAN: Module::Build loaded ok (v0.4003)

  CPAN.pm: Building D/DT/DTOWN/Net-SNMP-v6.0.1.tar.gz

Created MYMETA.yml and MYMETA.json
Creating new 'Build' script for 'Net-SNMP' version 'v6.0.1'
Building Net-SNMP
  DTOWN/Net-SNMP-v6.0.1.tar.gz
  C:\strawberry\perl\bin\perl.exe ./Build -- OK
Running Build test
t\ber.t .. ok
t\dsp.t .. ok
t\mp.t ... ok
t\usm.t .. ok
All tests successful.
```

Required Scripts continued....

2) Create a script “delayed_pdu_off.pl” using the sample code in the Appendix. Only the PDU IP addresses need to be changed [No other changes are required].



```
#! /usr/local/bin/perl

   use strict;
   use warnings;

   use Net::SNMP;

   my $OID_PDUdeviceCommand = '.1.3.6.1.4.1.318.1.1.12.3.1.1.0';

   my %host_data_off = (
      '10.216.252.197'   => '6',
      '10.216.252.198'   => '6',
   );

   for my $host (keys %host_data_off) {

      my ($session, $error) = Net::SNMP->session(
         -hostname     => $host,
         -community    => 'private',
         -nonblocking => 1,
      );

      if (!defined $session) {
         printf "ERROR: Failed to create session for host '%s': %s.\n",
                $host, $error;
         next;
      }

      my $result  = $session->get_request(
         -varbindlist => [ $OID_PDUdeviceCommand ],
         -callback     => [ \&get_callback, $host_data_off{$host} ],
      );

      if (!defined $result) {
         printf "ERROR: Failed to queue get request for host '%s': %s.\n",
```

Set the IP addresses of your PDU(S).

If you wish to use a 3rd party SNMP command line tool to issue the SET command to turn off the Rack PDU instead of the sample PERL script you may do so:

a)The OID for the SET command is .1.3.6.1.4.1.318.1.1.12.3.1.1.0
b) The syntax for the SET command is INTEGER, 6 (delayedAllOff)

3) Create a .cmd file which will execute the scripts to shutdown the Dell EqualLogic Storage Array and the Rack PDU after all the servers have been safely shut down or use the sample command file provided. The command file should contain the following content:

1. ping -n 180 127.0.0.1
2. cmd.exe /C perl C:\eql_script\eqlcliexec.pl -g <ip address> -a <login_name> -p <password> -f <shutdown_file>
3. ping -n 300 127.0.0.1
4. cmd.exe /C perl C:\DELL_SNMP\delayed_pdu_off.pl
5. <shutdown_file> is a text file e.g. C:\eql_script\shutdown.txt that must be present and contain the following lines:
   - ➢ Shutdown
   - ➢ Yes
6. Edit the delay values (seconds) marked in bold above to meet your requirements.

IMPORTANT: The values shown here are for illustration purposes only and may not be sufficient to safely shut down your servers and storage array. You should allow extra time for unforeseen circumstances . It is recommended to first test in a non-production environment to ensure that you have provided sufficient time for a safe shutdown.

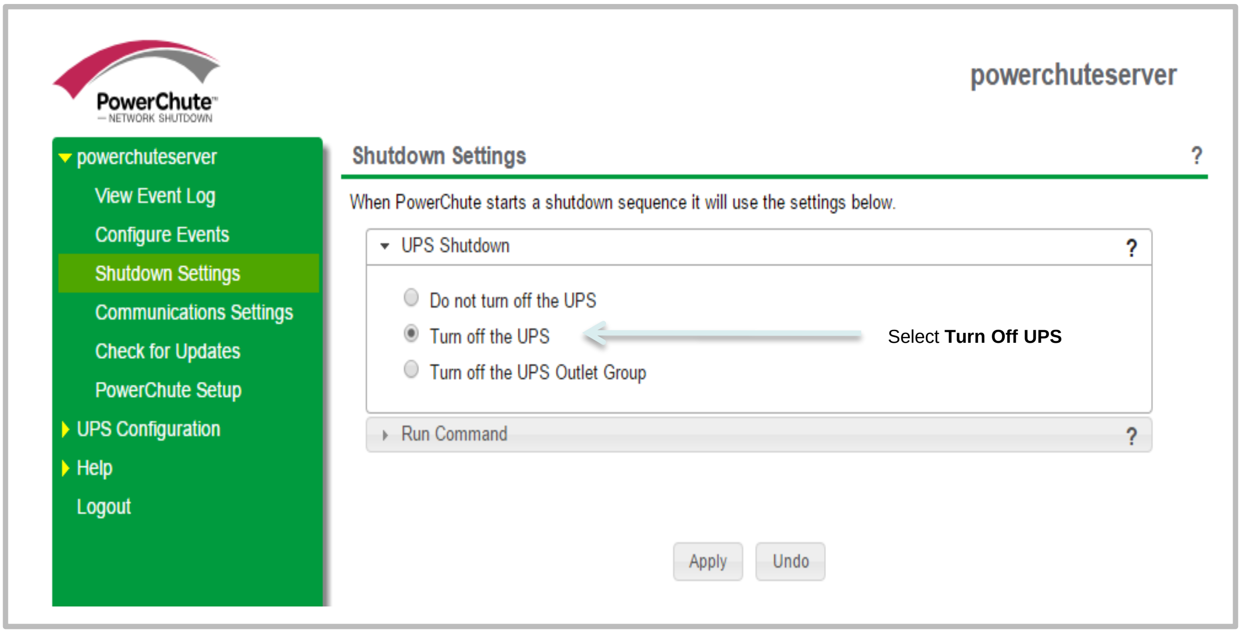
The 1st delay value is the time allowed for all Servers attached to the Storage Array to safely shut down before the script to shut down the Storage Array is executed.
The 2nd delay value is the time allowed for the Storage Array to safely shut down before the script to turn off the Rack PDU is executed.

NOTE: The sample script to turn off the PDU issues an SNMP set command which will start a delayed turn off of the PDU. Each outlet can be configured with a turn off delay value.

Configuration

PowerChute Settings

- The control server is the server which must be shut down last after all other servers have been shut down.
- This is also the server that runs the scripts to shut down the Storage Array and turn off the PDU.
- The command file to execute the scripts is configured on the PowerChute Configure Shutdown screen.
- The command file duration entered here should allow for all other servers and the storage array to shut down.
- It should correspond to the combined delay values you entered in the command file in the previous section.
- Ensure the value entered is long enough for all equipment powered by the PDU to shut down before the PDU.
- Only the PowerChute Agent on this control server is configured to turn off the UPS.



Shutdown Settings

When PowerChute starts a shutdown sequence it will use the settings below.

UPS Shutdown

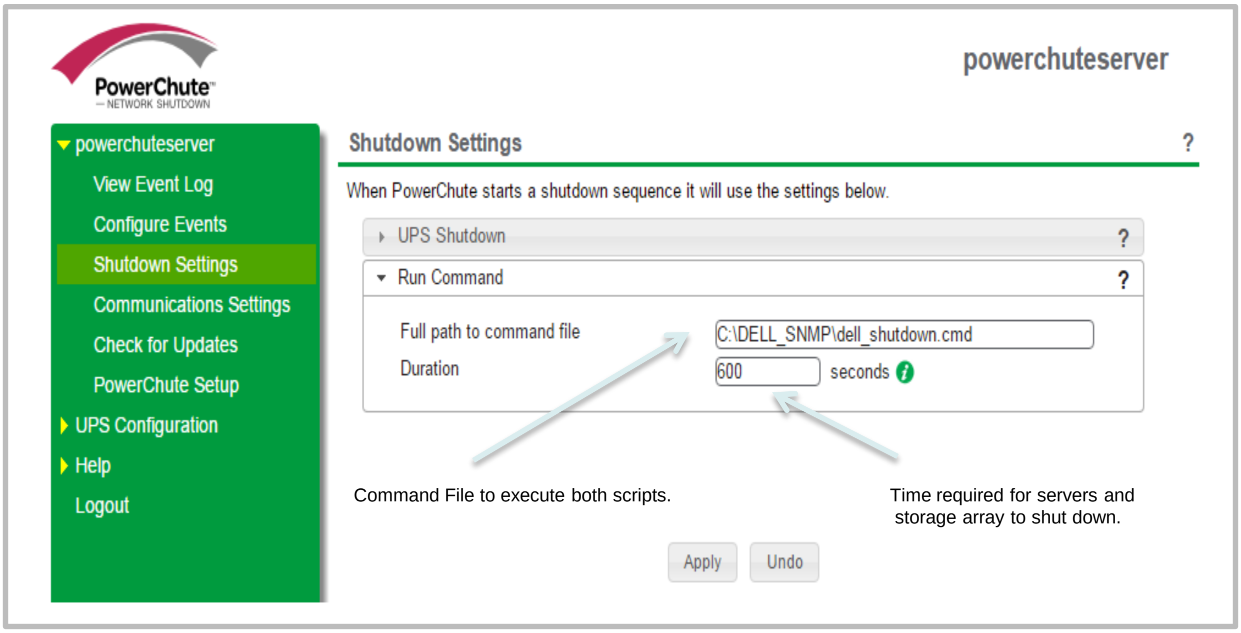
○ Do not turn off the UPS

● Turn off the UPS

○ Turn off the UPS Outlet Group

Run Command

Select **Turn Off UPS**



Shutdown Settings

When PowerChute starts a shutdown sequence it will use the settings below.

UPS Shutdown

Run Command

Full path to command file: C:\DELL_SNMP\dell_shutdown.cmd

Duration: 600 seconds

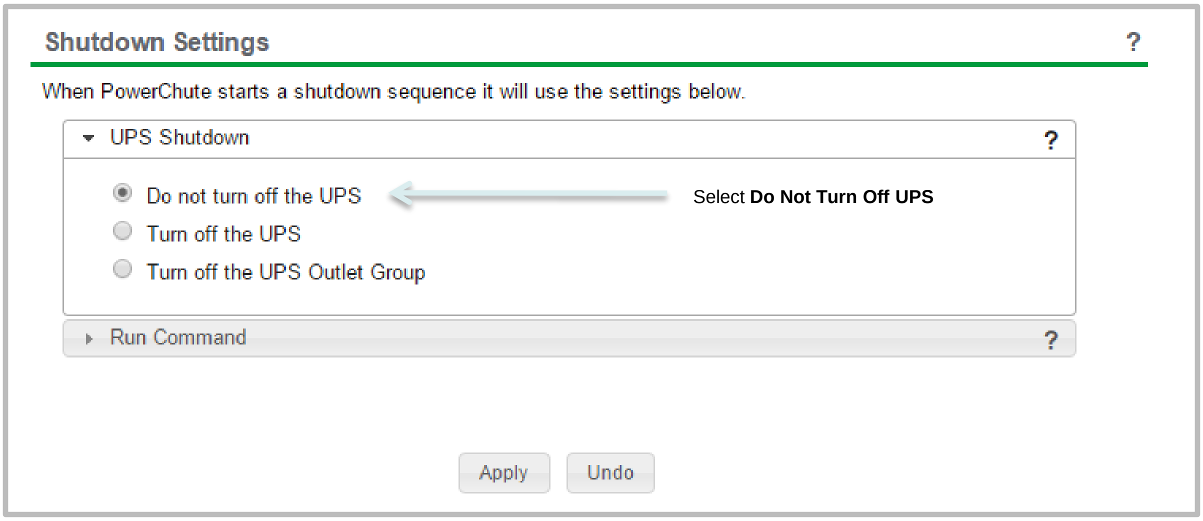
Command File to execute both scripts.

Time required for servers and storage array to shut down.

Configuration

PowerChute Settings

- The Configure Shutdown for all other PowerChute Agents has no command file configured and the UPS is not set to turn off.
  - These are the Agents on every other server other than the Control Server.



Shutdown Settings

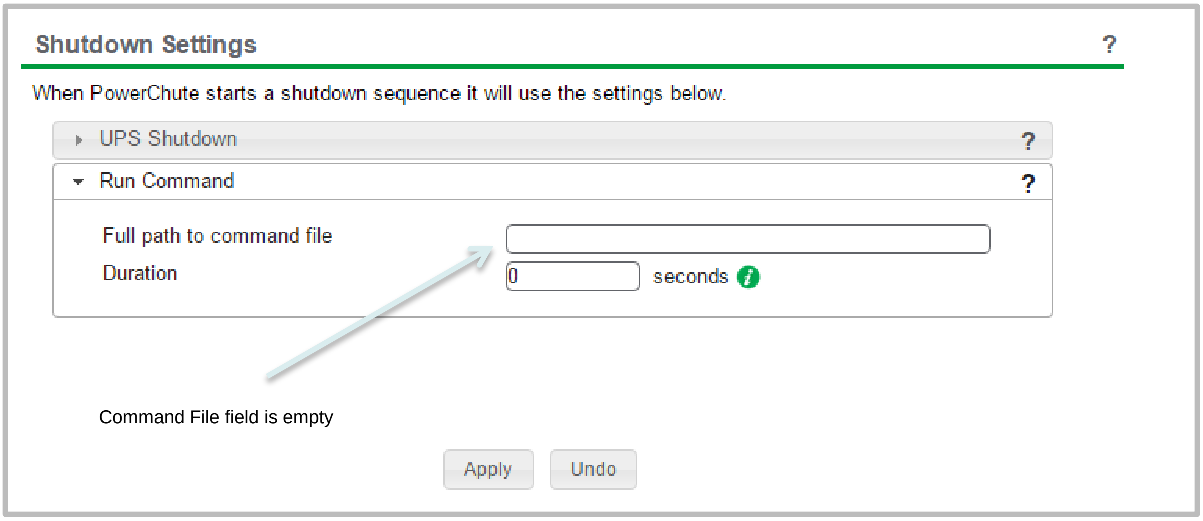
When PowerChute starts a shutdown sequence it will use the settings below.

UPS Shutdown

● Do not turn off the UPS
○ Turn off the UPS
○ Turn off the UPS Outlet Group

Run Command

Select **Do Not Turn Off UPS**



Shutdown Settings

When PowerChute starts a shutdown sequence it will use the settings below.

UPS Shutdown

Run Command

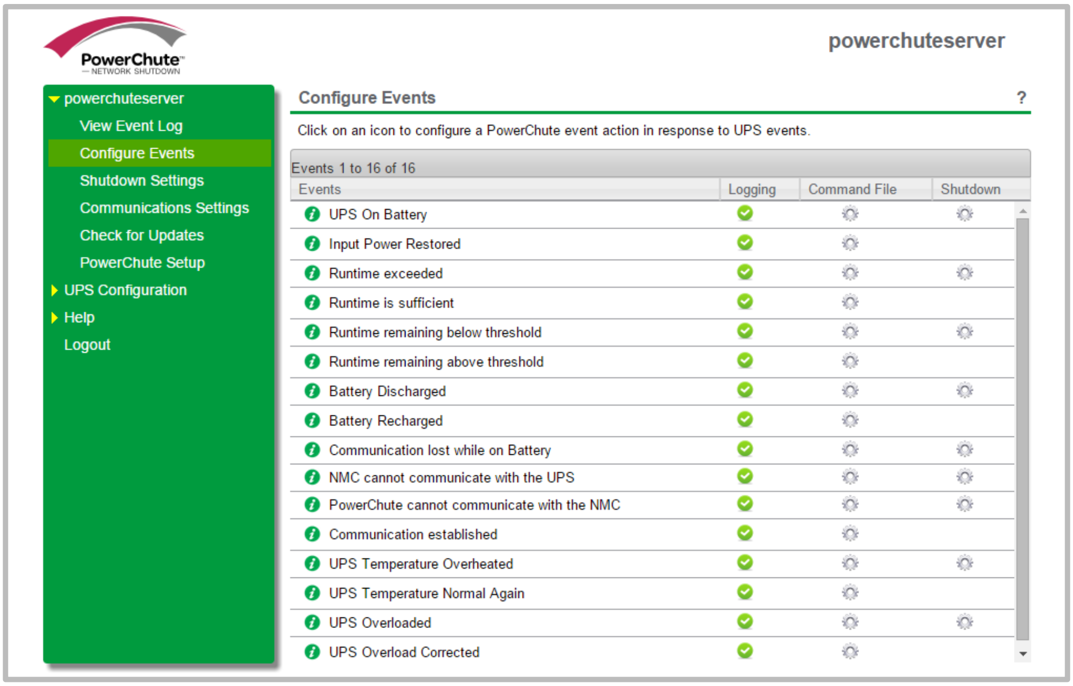
Full path to command file:
Duration: 0 seconds

Command File field is empty

Configuration

PowerChute Shutdown Delays

- Configure the events that you want PowerChute to shut down on for each server including the Control Server.
- The same shut down delay can be set for each event on all servers.



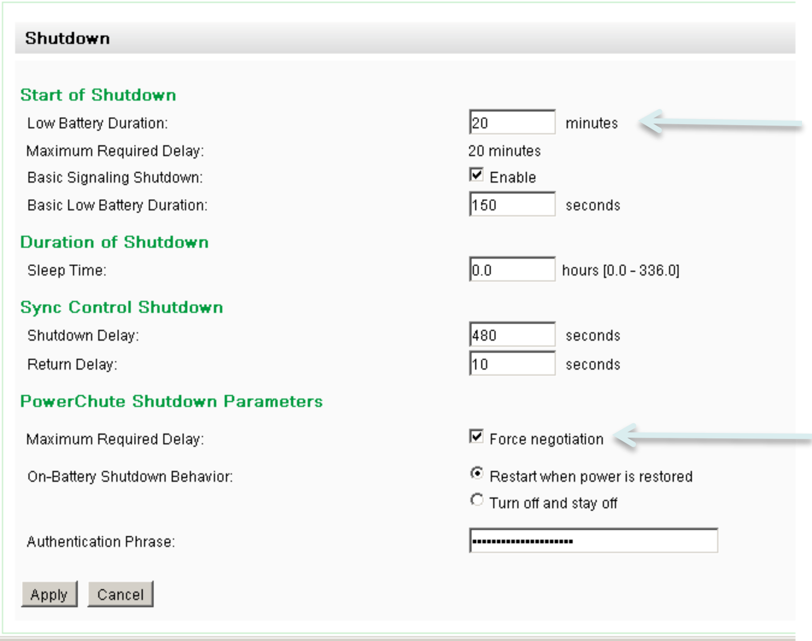
Configure Events

Click on an icon to configure a PowerChute event action in response to UPS events.

Events 1 to 16 of 16

Events	Logging	Command File	Shutdown
UPS On Battery	✓	⚙	⚙
Input Power Restored	✓	⚙	
Runtime exceeded	✓	⚙	⚙
Runtime is sufficient	✓	⚙	
Runtime remaining below threshold	✓	⚙	⚙
Runtime remaining above threshold	✓	⚙	
Battery Discharged	✓	⚙	⚙
Battery Recharged	✓	⚙	
Communication lost while on Battery	✓	⚙	⚙
NMC cannot communicate with the UPS	✓	⚙	⚙
PowerChute cannot communicate with the NMC	✓	⚙	⚙
Communication established	✓	⚙	
UPS Temperature Overheated	✓	⚙	⚙
UPS Temperature Normal Again	✓	⚙	
UPS Overloaded	✓	⚙	⚙
UPS Overload Corrected	✓	⚙	

- The UPS turn off sequence starts when the command file starts to execute.
- Ensure the UPS Low Battery duration is long enough for all servers, the storage array, PDU and control server to shut down.
- This can be configured via the Network Management Card user interface – Configuration – Shutdown tabs.



Shutdown

Start of Shutdown

Low Battery Duration: 20 minutes

Maximum Required Delay: 20 minutes

Basic Signaling Shutdown: ☑ Enable

Basic Low Battery Duration: 150 seconds

Duration of Shutdown

Sleep Time: 0.0 hours [0.0 - 336.0]

Sync Control Shutdown

Shutdown Delay: 480 seconds

Return Delay: 10 seconds

PowerChute Shutdown Parameters

Maximum Required Delay: ☑ Force negotiation

On-Battery Shutdown Behavior: ● Restart when power is restored ○ Turn off and stay off

Authentication Phrase: ••••••••••••••••••••••

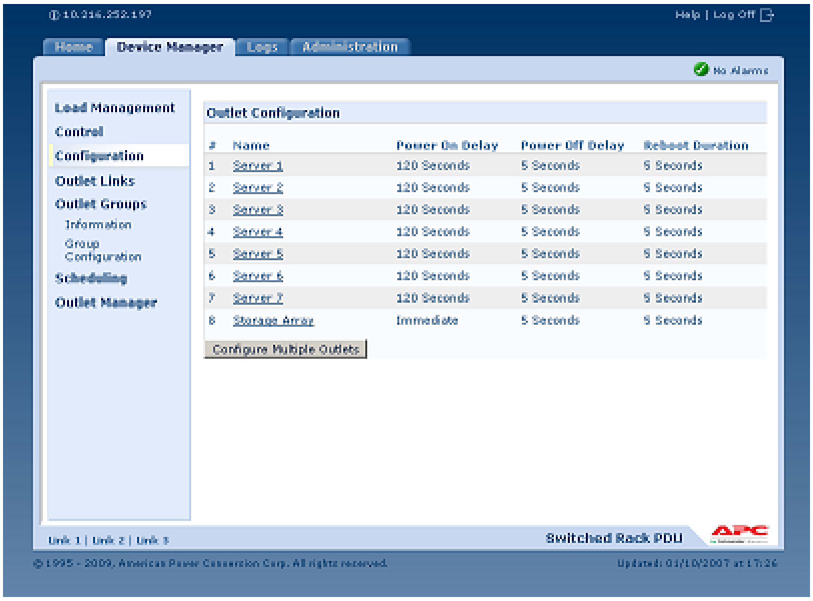
Ensure delay is long enough for all equipment to shut down safely before the UPS turns off

Must select Force Negotiation to ensure the delay is applied for all PowerChute Agents

Configuration

PDU Delays

- After the servers and storage array have been shut down, the command file will run the second script to power off the PDU.
- You need to configure the power off delay for each outlet on the PDU.
- In the screenshot below, all PDU outlets are configured to turn off 5 seconds after the second script is executed.
- This value is configurable but as all servers and the storage array are now off, it's recommended to turn the outlets off.



Outlet Configuration

#	Name	Power On Delay	Power Off Delay	Reboot Duration
1	Server 1	120 Seconds	5 Seconds	5 Seconds
2	Server 2	120 Seconds	5 Seconds	5 Seconds
3	Server 3	120 Seconds	5 Seconds	5 Seconds
4	Server 4	120 Seconds	5 Seconds	5 Seconds
5	Server 5	120 Seconds	5 Seconds	5 Seconds
6	Server 6	120 Seconds	5 Seconds	5 Seconds
7	Server 7	120 Seconds	5 Seconds	5 Seconds
8	Storage Array	Immediate	5 Seconds	5 Seconds

- When the UPS shutdown sequence completes, it will power off the PDU completely and the UPS will turn off.
- Once power is restored to the UPS, the PDU turns on and the outlets turn on according to the Power On Delays specified.
- The outlet that is powering the storage array should be configured to turn on immediately.
- The remaining outlets have a longer Power On Delay to allow enough time for the Storage Array to go online.
- This prevents other servers from coming online before the storage array.

Troubleshooting

In order to troubleshoot issues when the command file is executed please view cmdfile.log.

This file is located in C:\Program Files\APC\PowerChute\group1 folder and contains output from the command file and the perl scripts as they are executed.

The working directory when executing the shut down command file will always be C:\Program Files\APC\PowerChute\group1 - for this reason you should use the full path names where relevant.

Appendix

Sample Perl script to shut down rack PDU

Copy/paste the content below into a file named delay_pdu_off.pl. Only IP addresses (in red below) need to be edited.

```
#! /usr/local/bin/perl
use strict;
use warnings;
use Net::SNMP;
my $OID_PDUdeviceCommand = '.1.3.6.1.4.1.318.1.1.12.3.1.1.0';
 my %host_data_off = (
    '10.216.252.197'  => '6',
    '10.216.252.198'  => '6',
  );
  for my $host (keys %host_data_off) {
    my ($session, $error) = Net::SNMP->session(
      -hostname    => $host,
      -community   => 'private',
      -nonblocking => 1,
    );
    if (!defined $session) {
      printf "ERROR: Failed to create session for host '%s': %s.\n",
          $host, $error;
      next;
    }
my $result = $session->get_request(
      -varbindlist => [ $OID_PDUdeviceCommand ],
      -callback    => [ \&get_callback, $host_data_off{$host} ],
    );
    if (!defined $result) {
      printf "ERROR: Failed to queue get request for host '%s': %s.\n",
          $session->hostname(), $session->error();
    }
  }
  # Now initiate the SNMP message exchange.
 snmp_dispatcher();
 exit 0;
```

Sample Perl script to shut down rack PDU *continued...*

```
sub get_callback()
  {
    my ($session, $location) = @_;
    my $result = $session->var_bind_list();
    if (!defined $result) {
      printf "ERROR: Get request failed for host '%s': %s.\n",
             $session->hostname(), $session->error();
      return;
    }
    printf "The result for host '%s' is %s.\n",
           $session->hostname(), $result->{$OID_PDUdeviceCommand};
    $result = $session->set_request(
      -varbindlist =>
      [
        $OID_PDUdeviceCommand, INTEGER, $location,
      ],
      -callback    => \&set_callback,
    );
    if (!defined $result) {
      printf "ERROR: Failed to queue set request for host '%s': %s.\n",
             $session->hostname(), $session->error();
    }
    return;
  }
  sub set_callback
  {
    my ($session) = @_;
    my $result = $session->var_bind_list();
    if (defined $result) {
      printf "The result for host '%s' is '%s'.\n",
             $session->hostname(), $result->{$OID_PDUdeviceCommand};
    } else {
      printf "ERROR: Set request failed for host '%s': %s.\n",
             $session->hostname(), $session->error();
    }
    return;
}
```