GNU/Linux Administration - Feature #155

Configure APC UPS to Communicate with Ubuntu/Debian

08/01/2013 01:26 PM - Daniel Curtis

Status:	Closed	**Start date:**	08/01/2013
Priority:	Normal	**Due date:**	
Assignee:	Daniel Curtis	**% Done:**	100%
Category:		**Estimated time:**	1.00 hour
Target version:	Debian	**Spent time:**	1.00 hour

Description

APC make some great UPS products and they all have the ability to communicate with a host PC to advise of its state, eg: on mains, on battery, fault, etc. The support software is available aplenty for Windows and Mac, but what about Linux?

This guide will show you how to hook up an APC UPS to an Ubuntu/Debian Desktop or Server PC and allow your UPS to email you when mains power has failed, when mains power has been restored, and also give your PC ample opportunity to shutdown when battery on the UPS gets to a critical low.

Pre-requisites:

1. Ubuntu-based PC. I am using Ubuntu 10.04 Lucid Lynx in this guide.
2. An APC UPS. I'm using a Back-UPS RS 800 in this guide.
3. Details about your ISP's mail server, or your own local mail server.

Configuration:

Setup your APC UPS as outlined in the instruction manual. Ensure that the USB or serial cable is connected from the UPS to the PC. This is what the UPS uses to communicate with the PC.
If you are using Ubuntu Desktop, you can do this by going to the Applications menu, then Accessories and then Terminal. If you are using Ubuntu Server, simply login to your server by the console or SSH in remotely.

The Ubuntu and Debian repositories contain a UPS daemon specifically designed with APC UPS units in mind called APCUPSD, so let's install it:

```
sudo apt-get install apcupsd
```

Once installed, we need to configure it to suit your UPS. Bring up the configuration file by typing in:

```
sudo nano /etc/apcupsd/apcupsd.conf
```

In the editor, scroll down to UPSNAME and give the UPS a name, eg:

    UPSNAME MyUPS

Scroll down to UPSCABLE and change the parameter to the type of cable you are using to connect to your UPS. In my case, I use a USB cable that came with the UPS, so I changed this line to read:

    UPSCABLE usb

The next line to modify is the UPSTYPE line. Again, because I'm using USB I specify:

    UPSTYPE usb

If you have a DEVICE /dev/ttyS0 line after the UPSTYPE line, comment it out by adding a hash symbol at the start of the line like so:

    #DEVICE /dev/ttyS0

We do not need to modify anything else, so press CTRL+X, then "Y" and then Enter to save your changes and exit the Nano text editor.

Now we need to tell the system that the UPS daemon has been setup and is essentially ready to go. Type in:

```
sudo nano /etc/default/apcupsd
```

In the file that appears, modify the ISCONFIGURED line to say Yes", ie:

```
ISCONFIGURED=yes
```

Press CTRL+X, then "Y" and then Enter to save your changes and exit.

We're pretty much done here. Start the daemon now with the following command:

```
sudo service apcupsd restart
```

…or simply reboot your PC. Your PC is now monitoring the state of the UPS.

Let's check the UPS now. Type in the following command:

```
apcaccess
```

```
    APC     : 001,043,1045
    DATE    : Wed Jan 03 01:02:36 PST 2012
    HOSTNAME : hostname
    VERSION : 3.14.6  debian
    UPSNAME  : UPSNAME
    CABLE    : USB Cable
    MODEL   : Back-UPS BR  800
    UPSMODE  : Stand Alone
    STARTTIME:
    STATUS   : ONLINE
    LINEV    : 250.0 Volts
    LOADPCT  : 40.0 Percent Load Capacity
    BCHARGE  : 100.0 Percent
    TIMELEFT : 19.9 Minutes
    MBATTCHG : 5 Percent
    MINTIMEL : 3 Minutes
    MAXTIME  : 0 Seconds
    OUTPUTV  : 230.0 Volts
    SENSE    : Medium
    DWAKE    : 000 Seconds
    DSHUTD   : 000 Seconds
    LOTRANS  : 194.0 Volts
    HITRANS  : 264.0 Volts
    RETPCT   : 000.0 Percent
    ITEMP    : 29.2 C Internal
    ALARMDEL : Always
    BATTV    : 27.4 Volts
    LINEFREQ : 50.0 Hz
    LASTXFER : Low line voltage
    NUMXFERS : 0
    TONBATT  : 0 seconds
    CUMONBATT: 0 seconds
    XOFFBATT : N/A
    SELFTEST : NO
    STATFLAG : 0x07000008 Status Flag
    SERIALNO : xxxxxxxxxxxx
    BATTDATE : 2001-09-25
    NOMOUTV  : 230 Volts
    NOMINV   : 230 Volts
    NOMBATTV :  24.0 Volts
    NOMPOWER : 540 Watts
    FIRMWARE : 9.o5 .I USB FW:o5
    APCMODEL : Back-UPS BR  800
```

```
    END APC  :
```

Pat yourself on the back, you're basically done.

But hang on, what about setting up email notifications? Read on.

Setting up Email Notifications

By default, APCUPSD is configured to use sendmail to send emails, but sendmail is finicky. There is a better way using a similar application called sendemail instead.

Why use sendemail instead of sendmail? Well, sendemail is a simple command-line SMTP client. You construct a simple message with a recipient and off it goes, but the key difference here is that sendemail is a CLIENT, not a SERVER. This means for sendemail to work, you need a third-party mail server to send it through. You may already have a mail server on your network running Postfix or similar. If not, your ISP will most certainly have one that you can use.

Let's start by installing sendemail:

```
sudo apt-get install sendemail
```

- changeme – notifies you if the UPS battery needs changing.
- commfailure – notifies you if the PC loses communication with the UPS.
- commok – notifies you if lost communication is restored with the UPS.
- onbattery – notifies you if the UPS engages battery mode due to mains power failure.
- offbattery – notifies you if the UPS returns to mains mode after a mains power failure.

```
Now we need to configure apcupsd to use it. If you have a look inside the /etc/apcupsd directory,
you will see several important files as follows:
```

Let's modify the onbattery message:

```
sudo nano /etc/apcupsd/onbattery
```

You will notice there are some lines in this file that define some variables followed by a message and then finally a command that sends an email using the sendmail app. Since we are not using sendmail, we need to modify a number of areas. Replace the content of the entire text file with the following:

```
#!/bin/sh
#
# This shell script if placed in /etc/apcupsd
# will be called by /etc/apcupsd/apccontrol when the UPS
# goes on batteries.
#
SYSADMIN=jbloggs@mydomain.com.au
APCUPSD_MAIL="/usr/bin/sendemail"
HOSTNAME=`hostname`
MSG="Ubuntu PC Power Failure !!!"
#
(
echo " "
echo " ==================================="
echo " POWER FAILURE ON $HOSTNAME !!"
echo " ==================================="
echo " "
echo " The UPS on your Ubuntu PC has experienced a power problem that has required the UPS to
engage battery mode."
echo " "
echo " Do not panic! Remain calm..."
echo " "
echo "Current UPS status:"
echo " "
/sbin/apcaccess status
) | $APCUPSD_MAIL -u "$MSG" -f MyUbuntuPC@mydomain.com.au -t $SYSADMIN -s mail.myisp.com:25
```

```
    exit 0
```

So what have we done here? The first part of the file sets some variables, namely the email address(es) of the people required to be notified. If you need to send to more than one recipient, separate email addresses with a comma, eg:

    SYSADMIN=jbloggs@mydomain.com.au,fredsmith@thatotherdomain.com

We then define the path to the sendemail app, the hostname of the PC running the UPS daemon, a subject line for the email in question, and then a big email body explaining what's going on, in this case that the UPS battery mode has been engaged.

All this is then piped into the sendemail application with a fake sender's address to give you an idea where it came from and we also specify the mail server that the email will be sent via. In this case, we are using our ISP's mail server that has the address mail.myisp.com

Save your changes by pressing CTRL+X, then "Y" and then Enter.

You can now test the script by simply executing it (we don't have to disconnect the UPS' mains cable to trigger this until we're ready to test that). Execute the file using the command:

```
sh /etc/apcupsd/onbattery
```

Check for any errors and then check to see if you got an email in your mailbox advising that the UPS is on battery (the UPS of course is not, we're just testing the email).

If all is well, continue to change the other four message files to be similar to the above, changing the message as you go of course.

Pat yourself on the back – you now have custom, explicit messages to let you know what's going on with the UPS.

Testing

You can test your daemon setup easily enough by doing any of the following:

- Remove mains power cable from UPS.
- Remove the USB cable to the UPS.
- Verify that an email is sent for either of the error conditions above.

History

#1 - 05/09/2015 01:43 PM - Daniel Curtis

- Project changed from 15 to GNU/Linux Administration

- Target version set to Debian