

# APCUPSD: Customizing Event Handling

When `apcupsd` detects anomalies from your UPS device, it will make some decisions that usually result in one or more calls to the script located in `/etc/apcupsd/apccontrol`. The `apccontrol` file is a shell script that acts on the first argument that `apcupsd` passes to it. These actions are set up by default to sane behavior for all situations `apcupsd` is likely to detect from the UPS. However, you can change the `apccontrol` behavior for every single action.

To customize, so create a file with the same name as the action, which is passed as a command line argument. Put your script in the `/etc/apcupsd` directory.

These events are sent to the system log, optionally sent to the temporary events file (`/etc/apcupsd/apcupsd.events`), and they also generate a call to `/etc/apcupsd/apccontrol` which in turn will call any scripts you have placed in the `/etc/apcupsd` directory.

Normally, `/etc/apcupsd/apccontrol` is called only by `apcupsd`. Consequently, you should not invoke it directly. However, it is important to understand how it functions, and in some cases, you may want to change the messages that it prints using `wall`. We recommend that you do so by writing your own script to be invoked by `apccontrol` rather than by modifying `apccontrol` directly. This makes it easier for you to upgrade to the next version of `apcupsd`.

In other case, you may want to write your own shell scripts that will be invoked by `apccontrol`. For example, when a power fail occurs, you may want to send an email message to `root`.

To write your own routine for the `powerout` action, you create shell script named `powerout` and put it in the `lib` directory (normally `/etc/apcupsd`). When the `powerout` action is invoked by `apcupsd`, `apccontrol` will first give control to your script. If you want `apccontrol` to continue with the default action, simply exit your script with an exit status of zero. If you do not want `apccontrol` to continue with the default action, your script should exit with the special exit code of 99. However, in this case, please be aware that you must ensure proper shutdown of your machine if necessary.

Some sample scripts (`onbattery` and `mainsback`) that email power failure messages can be found in `/etc/apcupsd` after an install or in the `platforms/etc` directory of the source code.

## **apccontrol Command Line Options**

When `apcupsd` detects an event, it calls the `apccontrol` script with four arguments as:

```
apccontrol event ups-name connected powered
```

where:

- event*  
is the event that occurred and it may be any one of the values described in the next section.
- ups-name*  
is the name of the UPS as specified in the configuration file (not the name in the EEPROM).
- connected*  
is 1 if `apcupsd` is connected to the UPS via a serial port (or a USB port). In most configurations, this will be the case. In the case of a Slave machine where `apcupsd` is not directly connected to the UPS, this value will be 0.
- powered*  
is 1 if the computer on which `apcupsd` is running is powered by the UPS and 0 if not. At the moment, this value is unimplemented and always 0.

The following *event* names are supported:

### **annoyme**

When a shutdown is scheduled, and the time specified on the ANNOYME directive in the apcupsd.conf file expires, this event is generated.

*Default:* wall a message

### **changeme**

When apcupsd detects that the mains are on, but the battery is not functioning correctly, this event is generated. It is repeated every x hours.

*Default:* wall a message

### **commfailure**

This event is generated each time the communications line with the computer is severed. This event is not detected on dumb signaling UPSes.

*Default:* wall a message

### **commok**

After a commfailure event is issued, when the communications to the computer is re-established, this event will be generated.

*Default:* wall a message

### **doreboot**

This event is depreciated and should not be used.

*Default:* Shuts down the system using shutdown -h or similar

### **doshutdown**

When the UPS is running on batteries and one of the limits expires (time, run, load), this event is generated to cause the machine to shutdown.

*Default:* Shuts down the system using shutdown -h or similar

### **emergency**

Called for an emergency system shutdown. (What triggers such a shutdown is unclear...) After completing this event, apcupsd will immediately initiate a doshutdown event.

*Default:* wall a message

### **failing**

This event is generated when the UPS is running on batteries and the battery power is exhausted. The event following this one will be a shutdown.

*Default:* wall a message

### **loadlimit**

This event is generated when the battery charge is below the low limit specified in the apcupsd.conf file. After completing this event, apcupsd will immediately initiate a doshutdown event.

*Default:* wall a message

## **powerout**

This event is generated immediately when apcupsd detects that the UPS has switched to batteries. It may be due to a short powerfailure, an automatic selftest of the UPS, or a longer powerfailure.

*Default:* wall a message

## **onbattery**

This event is generated 5 or 6 seconds after an initial powerfailure is detected. It means that apcupsd definitely considers the UPS to be on batteries. The onset of this event can be delayed by the ONBATTERYDELAY apcupsd.conf configuration directive.

*Default:* wall a message

## **offbattery**

This event is generated when the mains return only if the onbattery event has been generated.

*Default:* wall a message

## **mainsback**

This event is generated when the mains power returns after a powerout condition. The shutdown event may or may not have been generated depending on the parameters you have defined and the length of the power outage.

*Default:* nothing

## **remotedown**

This event is generated on a slave machine when it detects either that the master has shutdown, or that a onbattery situation exists and the communications line has been severed.

*Default:* wall a message

## **runlimit**

This event is generated when the MINUTES value defined in the apcupsd.conf file expires while in a power fail condition. The MINUTES is the remaining runtime as internally calculated by the UPS and monitored by apcupsd. After completing this event, apcupsd will immediately initiate a doshutdown event.

*Default:* wall a message

## **timeout**

This event is generated when the TIMEOUT value defined in the apcupsd.conf file expires while in a power fail condition. It indicates that the total time in a power failure has been exceeded and the machine should be shutdown. After completing this event, apcupsd will immediately initiate a doshutdown event.

*Default:* wall a message

## **startselftest**

This event is generated when apcupsd detects a self test by the UPS. Normally due to the 6 second onbattery delay default time, self test events are not detected.

*Default:* nothing

**endselftest**

This event is generated when the end of a self test is detected.

*Default:* nothing

**battdetach**

This event is generated when apcupsd detects that the UPS battery has been disconnected.

*Default:* nothing

**battattach**

This event is generated when apcupsd detects that the UPS battery has been reconnected after a battdetach event.

*Default:* nothing