

Chapter 15: Automatic UPS Product Startup and Shutdown

This chapter covers configuring your system to support automatic startup and shutdown of **UPS** products, and installing individual **UPS** product instances to start and stop automatically. The current bootstrap procedure (see Chapter 14: *Installing UPS and UPD from Bootstrap*) ensures that when **UPS** gets installed on a system, it is configured to enable this feature.

Note that very few products need to be run automatically; a couple of examples are **juke** and **apache**.

15.1 Configuring Your Machine to Allow Automatic Startup/Shutdown

Two scripts are supplied by **UPS** and used to run products automatically at boot time, `ups_startup` and `ups_shutdown`. A third script, called `ups`, must be supplied by you and placed in the `init.d` directory where it will be executed at boot time. It is used to configure your machine for automatic startup/shutdown and to call the first two scripts. When **UPS** gets installed and configured on a system, `ups_startup` and `ups_shutdown` get copied into separate directories under `$PRODUCTS/.upsfiles`, as follows:

```
$PRODUCTS/.upsfiles/startup/ups_startup
```

```
$PRODUCTS/.upsfiles/shutdown/ups_shutdown
```

We encourage you to use the sample `ups` script given below, with no changes except the database path (defined by `upsdb`). It works for all supported UNIX flavors.

The ups Script Particulars

The script must be called `ups`. The location of the `init.d` directory in which it must reside is OS-specific, as follows:

Operating System	Directory
IRIX, SunOS	<code>/etc/init.d</code>
Linux	<code>/etc/rc.d/init.d</code>
OSF1 ^a	<code>/sbin/init.d</code>

a. On OSF1 systems the system start-up directories `init.d`, `rc2.d` and `rc0.d` are under `/sbin`, not `/etc`.

Set the `ups` file ownership and permissions properly by running:

```
% chown 0 ups
```

```
% chgrp 0 ups
```

The ups Script Contents

```
#!/bin/sh

upsdb=/local/ups/db
state=$1
case $state in
  'start')
    start=$upsdb/.upsfiles/startup/ups_startup
    (while [ ! -f $start ]; do sleep 5; done; $start) &
    ;;
  'stop')
    $upsdb/.upsfiles/shutdown/ups_shutdown
    ;;
  'config')
    case $0 in
      /*) initd=$0;;
      *) initd='pwd'/$0;;
    esac
    sfile='echo $initd | sed -e 's;init.d;/rc3.d/S99;''
    kfile='echo $initd | sed -e 's;init.d;/rc0.d/K01;''
    ln -s $initd $sfile
    ln -s $initd $kfile
    sfile='echo $initd | sed -e 's;init.d;/rc5.d/S99;''
```

```

        kfile=`echo $initd | sed -e 's;init.d;/rc6.d/K01;''
        ln -s $initd $sfile
        ln -s $initd $kfile
        ;;
*)
        echo "usage: $0 {start|stop|config}"
        ;;
esac

```

15.2 Installing a UPS Product to Start and/or Stop Automatically

This section contains the information you need in order to install appropriate **UPS** products to run automatically. A rudimentary understanding of actions and functions in table files is helpful (see Chapter 34: *Actions and ACTION Keyword Values* and Chapter 35: *Functions used in Actions*). The autostart/autostop processes are run via a set of control files and the commands **ups start** and **ups stop**.

15.2.1 Determine if Auto Start/Stop Feature is Enabled

Unless you're installing the product **UPS** itself, you don't need to understand how the automatic startup/shutdown feature gets enabled, but you may need to determine whether it is enabled or not. The files `$PRODUCTS/.upsfiles/startup/ups_startup` and `$PRODUCTS/.upsfiles/shutdown/ups_shutdown` are the scripts that initiate the startup/shutdown functions in **UPS**. The automatic startup/shutdown feature is enabled if and only if these three conditions are met:

- the `ups_startup` and `ups_shutdown` files exist
- the appropriate system startup files on your machine are configured to call these files (described in section 15.1 *Configuring Your Machine to Allow Automatic Startup/Shutdown*)
- an appropriate control file exists

15.2.2 Determine if Product is Appropriate for Autostart

Products that are appropriate to run in this fashion (should) come equipped with **START** and **STOP** actions in their table files. For products configured in the old **UPS** style (prior to v4), the functions comprising these actions will

probably be `sourceRequired(/path/to/<start_script>)` and `sourceRequired(/path/to/<stop_script>)`, respectively. This function is described in section 35.3.31 *sourceRequired*. The specified paths must point to executables that contain start and stop instructions for the product.

Often these products also come with an TAILOR action in the table file (see section 4.6.2 *Tailoring a Product*). Once the product has been configured and tailored properly, ACTION=START functions are run at boot time to start the product and ACTION=STOP functions are run at system shutdown to stop it.

15.2.3 Edit Control File(s)

In order to make known to the system that your product is to be started at boot time, you will need to add a specific line of text to the appropriate control file. This line provides the actual start command for the product. Go to the `$PRODUCTS/.upsfiles/startup` directory¹. There you may find one or more files with the name `<node>.products` (where `<node>` is one of the nodes in your cluster, e.g., `fsgi02.products`) and/or `<flavor>.products` (where `<flavor>` is one of the flavors in your cluster, e.g., `IRIX+5.products`). If you want your product to run as an automatic startup process on a single node in your cluster, edit the file appropriate to that node, or create the file if it doesn't exist. If you want it to run on all nodes of a particular flavor, edit or create the file appropriate to that flavor. A line for a particular product can exist in more than one file, i.e., both the corresponding node-specific and the flavor-specific files. The system runs all these files at boot time and uses only the first `ups start` command it finds for a product.

Add a line of the following format to the appropriate file(s):

```
/bin/su - <login_id> -c "ups start [<options>] <product>
 [<version>]"
```

Notes:

- The login id is that under which `ups start` must be run. If the login id is `root`, you can leave out the portion `/bin/su - <login_id> -c`, and the line can start from `ups start` (no quotes needed in this case).
- The `-c` shown here does not refer to the current chain, rather it is an option to the `/bin/su` call which tells it to execute `ups start`.
- The `ups start` script should initialize the **UPS** environment (see section 2.7.1 *Initializing the UPS Environment*), and the login id used here should not. You could opt to have the login id initialize the environment



1. If `$PRODUCTS` includes more than one database, use the `startup` directory in the database in which the active (usually current) instance of **UPS** resides.

instead, but we recommend against it, especially if it's the *root* account. If the *root* account runs the standard Fermi files, then you can't use it to get into a system where there are problems with the Fermi/UPS environment.

In the directory `$PRODUCTS/.upsfiles/shutdown` you will find files named similarly to the startup control files. The files here tell your system to stop the process at shutdown. You will need to edit one or more of them. Add a line of the following format to the appropriate file(s):

```
/bin/su - <login_id> -c "ups stop [<options>] <product>
[<version>]"
```



The same notes regarding the login id and `-c` option apply.

Make sure the permissions of all the `<node>.products` and `<flavor>.products` files are set to 744. This ensures that the files will be executable by *root* and that they have appropriate permissions for avoidance of security holes. If your products area is NFS mounted to all the appropriate machines (i.e., common to them), you only need to create these files once. If not, you need to create these files once for each products area.

15.2.4 Summary

- 1) Declare the product to the **UPS** database (if configuration via **ups configure** is required it gets done in this step by default; see section 4.6.1 *Configuring a Product*).
- 2) Tailor the product (usually but not always required).
- 3) Add a **ups start** control line to the appropriate file(s) in the `$PRODUCTS/.upsfiles/startup` directory.
- 4) Add a **ups stop** control line to the appropriate file(s) in the `$PRODUCTS/.upsfiles/shutdown` directory.

When the system is restarted, your process should start running on the nodes you've designated.

15.3 Disabling UPS Automatic Start/Stop of Processes

For a Single Product

To disable automatic start and stop for a single product, just remove or comment out the corresponding lines in the `<node>.products` and `<flavor>.products` files.

Disable Feature in UPS

To disable the **UPS** automatic start and stop mechanism at boot time and shutdown:

- Remove (or rename) the file `ups_startup` from the `$PRODUCTS/.upsfiles/startup` directory and the file `ups_shutdown` from the `$PRODUCTS/.upsfiles/shutdown` directory.
- Remove (or rename) all links as set in the `ups` script (e.g., `/etc/rc2.d/S99ups` and `/etc/rc0.d/K01ups` for IRIX) to the OS-specific script in the system boot area that calls the `ups_startup` and `ups_shutdown` scripts.
- Edit, remove or rename the above-mentioned OS-specific script (e.g., `/etc/init.d/ups` for IRIX).
- Empty, remove or rename the `$PRODUCTS/.upsfiles/startup` and `$PRODUCTS/.upsfiles/shutdown` directories.

15.4 A Summary of the UPS Automatic Start-up Process

Since so many different files and directories have similar names, it can be difficult to keep track of the role each plays. The process which takes place at system start-up when automatic start and stop are enabled can be summarized as follows:

- 1) At boot time, the link in the system's start-up area (e.g., `/etc/rc3.d/S99ups` for SunOS) points to the `ups` file (in the directory appropriate for the flavor) which runs `/path/to/ups_database/.upsfiles/startup/ups_startup`.
- 2) `$PRODUCTS/.upsfiles/startup/ups_startup` runs the appropriate `$PRODUCTS/.upsfiles/startup/<node>.products` and/or `$PRODUCTS/.upsfiles/startup/<flavor>.products`. It also runs `setups.[c]sh` so that appropriate environment variables and aliases get set.
- 3) The `<node>` or `<flavor>.products` file in turn runs the **ups start** command.
- 4) **ups start** executes the **START** action in the table file of the product.

The process for automatic stop is similar.

