Appendix J. Automatic UPS Product Startup and Shutdown

This appendix covers configuring your system to support automatic startup and shutdown of **UPS** products, and installing individual **UPS** product instances to start and stop automatically.

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

J.1 OS-Dependent Preparatory Procedures

You must configure your **UPS** user node to look for and execute the `.upsfiles/startup/ups_startup` and `.upsfiles/shutdown/ups_shutdown` files maintained in the **UPS** database (e.g., `$PRODUCTS/.upsfiles...`)

J.1.1 Procedure for AIX Systems

On AIX systems, boot time and shutdown procedures are initiated through the **init** process which is controlled by the `/etc/inittab` file. The following steps allow you to connect the **UPS** start and stop procedures to this AIX-standard control file:

1) Add to the *top* of the `/etc/inittab` file:

```
rcups2:0:wait:/etc/ups.shutdown
```

This causes **UPS** to take control at the *beginning* of the shutdown process.

2) Add to the *bottom* of the same file:

```
rcups:2:wait:/etc/rc.ups
```

This causes **UPS** to take control at the *end* of the boot process.

3) Create the file `/etc/ups.shutdown` and enter as the contents the path to your `$PRODUCTS/.upsfiles/shutdown/ups_shutdown` file. For example:

```
/path/to/ups_database/.upsfiles/shutdown/ups_shutdown
```

4) Create the file `/etc/rc.ups` and enter as the contents the path to your `$PRODUCTS/.upsfiles/startup/ups_startup` file. For example:

```
/path/to/ups_database/.upsfiles/startup/ups_startup
```

5) For these last two files, change the permission to 744, the user ownership to *root* and the group ownership to *sys*.

J.1.2 Procedure for IRIX Systems

1) In the `/etc/init.d` directory, create and edit a file called `ups`. Enter the following text into the file:

```
state=$1
case $state in
  'start')
      /path/to/ups_database/.upsfiles/startup/ups_startup
      ;;
  'stop')
     /path/to/ups_database/.upsfiles/shutdown/ups_shutdown
     ;;
  'config')
     ln -s /etc/init.d/ups /etc/rc2.d/S99ups
     ln -s /etc/init.d/ups /etc/rc0.d/K01ups
     ;;
  *)
     echo "usage: $0 {start|stop}"
     ;;
esac
```

2) Change the permission on `/etc/init.d/ups` to 744, the user ownership to *root* and the group ownership to *sys*.

J.1.3 Procedure for SunOS Systems

1) In the `/etc/init.d` directory, create and edit a file called `ups`. Enter the following text into the file:

```
state=$1
case $state in
  'start')
      /path/to/ups_database/.upsfiles/startup/ups_startup
      ;;
  'stop')
     /path/to/ups_database/.upsfiles/shutdown/ups_shutdown
     ;;
  'config')
     ln -s /etc/init.d/ups /etc/rc3.d/S99ups
     ln -s /etc/init.d/ups /etc/rc0.d/K01ups
     ;;
  *)
     echo "usage: $0 {start|stop}"
     ;;
esac
```

2) Change the permission on `/etc/init.d/ups` to 744, the user ownership to *root* and the group ownership to *staff*.

J.1.4 Procedure for OSF1 Systems

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

1) In the `/sbin/init.d` directory, create and edit a file called `ups`. Enter the following text into the file:

```
state=$1
case $state in
  'start')
      /path/to/ups_database/.upsfiles/startup/ups_startup
      ;;
  'stop')
     /path/to/ups_database/.upsfiles/shutdown/ups_shutdown
     ;;
```

```
  'config')
     ln -s /sbin/init.d/ups /etc/rc2.d/S99ups
     ln -s /sbin/init.d/ups /etc/rc0.d/K01ups
     ;;
  *)
     echo "usage: $0 {start|stop}"
     ;;
esac
```

2) Change the permission on `/sbin/init.d/ups` to 744, the user ownership to *root* and the group ownership to *staff*.

J.2 Install UPS to Allow Automatic Startup/Shutdown

This topic is currently covered in the on-line document located at `http://www.fnal.gov/docs/products/ups`.

J.3 Install a UPS Product to Start and/or Stop Automatically

UPS can be configured to allow automatic startup of appropriately configured **UPS** products at system boot time and automatic stop at system shutdown. Unless you're installing the product **UPS** itself, you don't need to understand how this feature gets enabled, but you may need to determine whether it is enabled or not. This section assumes that you understand the concepts of actions and functions in table files.

The files `.upsfiles/startup/ups_startup` and `.upsfiles/shutdown/ups_shutdown` (where `.upsfiles` is located in the **UPS** database containing the active instance of **UPS**) 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:

- these files exist
- the appropriate system startup files on your machine are configured to call these files (see section J.1)
- an appropriate control file (described below) exists

The information you need in order to install **UPS** products to run automatically in an appropriately configured **UPS** implementation follows. This feature is controlled via a set of control files and the commands **`ups start`** and **`ups stop`**.

Product start and stop Files

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(`*****start_script_path*****`)`** and **`sourceRequired(`*****stop_script_path*****`)`**, respectively. These 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 *Tailoring* under section 4.3). 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.

UPS Process Control Files

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` directory[1]. 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**.
- For OSF1 v4.0b systems, leave off the **-c** flag; it assumes what follows is a command.
- The `ups start` script should initialize the **UPS** environment (see section 2.4.1), and the ***login_id*** used here should not. You could opt to have the ***login_id*** initialize the environment 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.

Similarly, to tell your system to stop the process at shutdown, go to the directory `$PRODUCTS/.upsfiles/shutdown`. There you will find similarly named files, one or more of which you will need to edit. 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 mode of all the `{node}.products` and `{flavor}.products` files is 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.

Summary for Implementing an Automatic Process under UPS Control

1) Declare the product to the **UPS** database (if configuration is required it gets done in this step by default; see section 4.3).

2) Tailor the product (usually required).

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.

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.

J.4 Disable UPS Automatic Start/Stop of Processes

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

- remove the file ups_startup from the $PRODUCTS/.upsfiles/startup directory and the file ups_shutdown from the $PRODUCTS/.upsfiles/shutdown directory
- remove all links (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 or remove the above-mentioned OS-specific script (e.g., /etc/init.d/ups for IRIX)
- empty or remove the $PRODUCTS/.upsfiles/startup and $PRODUCTS/.upsfiles/shutdown directories

To disable automatic start and stop for a single product, just remove the corresponding lines in the {node}.products and {flavor}.products files.

J.5 A Summary of the 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) IRIX, SunOS and OSF1: At boot time, the link in the system's start-up area (e.g., /etc/rc3.d/S99ups for SunOS) points to the /etc/init.d/ups file (IRIX and SunOS) or the /sbin/init.d/ups file (OSF1) which runs /path/to/ups_database/.upsfiles/startup/ups_startup.

   AIX: At boot time, the /etc/inittab file calls /etc/rc.ups which in turn calls /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.

J.6 Sample start and stop Scripts

We'll use scripts for **juke** v5_2 as examples.

start

```
#!/bin/sh

case “$0” in
/*)     JUKE_DIR=`echo $0 | sed -e ‘s;/ups/start;;’`
        export JUKE_DIR
        PATH=$JUKE_DIR/bin:$PATH
        ;;
*)      ;;
esac

cd $JUKE_DIR/log

host=`hostname`
local=”`$JUKE_DIR/bin/juke show jukebox | grep $host | sed -e ‘s/@.*//’`”

if [ “$local” != ““ ]
then
        if [ -f $JUKE_DIR/log/jukerpcd.$host.pid ]
        then
                # it looks like one is running
                if kill -0 `cat $JUKE_DIR/log/jukerpcd.$host.pid`
                then
                        #daemon is already running, we’re done
                        exit 0
                fi
        fi
        nohup $JUKE_DIR/bin/jukerpcd >> jukerpcd.$host.out 2>&1 </dev/null &
        echo $! > $JUKE_DIR/log/jukerpcd.$host.pid

        sleep 10        # wait for jukerpcd to wake up

        for i in $local
        do
                if [ “`uname -s`” = “AIX” ]
                then
                        # AIX driver doesnt autoconfigure, so configure it
                        dev=`$JUKE_DIR/bin/juke show jukebox |
                                grep $local |
                                sed -e ‘s;.*/dev/;;’ -e ‘s/[    ].*//’`
                        mkdev -l $dev
                fi
                $JUKE_DIR/bin/juke online -j $i &
        done
fi
```

stop

```
#!/bin/sh

if [ ““ = “$JUKE_DIR” ]
then
        JUKE_DIR=`echo $0 | sed -e ‘s;/ups/stop;;’`
        export JUKE_DIR
        PATH=$JUKE_DIR/bin:$PATH
fi


host=`hostname`
```

```
if [ -f $JUKE_DIR/log/jukerpcd.$host.pid ]
then
        kill -15 ‘cat $JUKE_DIR/log/jukerpcd.$host.pid‘
        rm $JUKE_DIR/log/jukerpcd.$host.pid
fi
```