

Suggestions on Using CRON

Are you using cron, the UNIX utility for repetitive background jobs? Many sites use it only for automating their backups. Because file maintenance (as well as system maintenance) is a repetitive job, cron offers a great deal of utility once you "break the code" on how to get it to work. I think an example is worth a lot. Several of our recently installed FAST clients have asked for examples of how to setup FAST in cron. In the following paragraphs, we will demonstrate the way we do it. Because we have clients running PI/Open, uniVerse and UniData, we wanted a technique which would work for all three databases.

The first step is to determine what you want to run and when. We run our "Gather Statistics and Type Analysis" job every Sunday night at 2am. Step two is to create the uniVerse, Unidata or PI/Open paragraph you wish to run. In FAST.CONTROL we have a paragraph to perform the file analysis and typing. Here is a sample:

(our record name is RESIZE.SETUP in FAST.CONTROL)

```
PA
LOGTO /users/peggy
FAST.GATHER.STATS (BATCH) (NAMES)
FAST.TYPE.ANALYSIS (BATCH) (NAMES)
LOGTO /users/fl_data
FAST.GATHER.STATS (BATCH) (NAMES)
FAST.TYPE.ANALYSIS (BATCH) (NAMES)
```

In the fast directory, we have two UNIX files which are used to execute the FAST paragraph called RESIZE.SETUP. One is a file of commands to be executed in uniVerse (or PI/Open or UniData.) The second is a UNIX script. We call our command file "resize.setup" (in lower case.) This is just our convention, you may name yours what you wish. Here is an example of the contents of this file:

FAST.BATCH RESIZE.SETUP (INIT)

NOTE: This statement is the command to submit a FAST BATCH job. This statement can be any command that you can enter at TCL such as a paragraph or PROC name. This technique is not limited to FAST!

The UNIX script contains the commands to setup the UNIX PATH so that the cron job will have the PATH to ../uv/bin (or to the PI/Open or UniData bin subdirectories). This is a key step. The script also runs uniVerse using the command file (resize.setup above) as input.

Here is an example of our script which we call run.setup. Note that we use lower case for this script name which is one of our naming conventions.

```
PATH=$PATH:/u1/uv/bin
export PATH
uv < resize.setup
```

Remember to change permissions to allow execution of these two files. We use the following command:

```
chmod 777 resize.setup run.setup
```

It's a very good idea to test your script at this point. To execute it, get to the UNIX prompt in the fast directory and enter the following command:

```
# ./run.setup
```

This test will ensure that the setup you've created is doing what you want. The FAST.CONTROL paragraph is executed by uniVerse using the "resize.setup" command file. But, we need the "run.setup" script to establish the PATH variable and to pass our command file into uniVerse (or UniData or PI/OPEN). After you have tested all of the scripts and paragraphs (and know they work) you are ready to schedule your cron job. Most UNIX systems provide a "front end" for using cron such as SMIT on the RS6000 and SAM on HP. Or you can use "vi" to build your crontab. On our HP using "sam" we entered the command:

```
(cd /u1/fast;/bin/sh run.setup) > /u1/fast/cron.out 2>&1
```

Because we can't count on cron to have environment variables and pathnames set, we must explicitly define everything we want done. The first command is "cd /u1/fast". We want FAST to run from the fast directory. The next command "/bin/sh" invokes the UNIX shell to execute our script "run.setup". The ">" redirects standard out into a file in the fast directory, "cron.out". The "2>&1" redirects standard error to the same file.

Please be VERY careful when working with cron. Look at your system's documentation for cron. It is very easy to overwrite an existing entry in the "crontabs" directory which I have done on several occasions. Here is a technique which I have tested on our HP and it worked for me.

First, to use FAST we strongly recommend that you be a superuser. Most sites use the "root" login. Does "root" have an entry in the crontabs directory on your system? To check the existence of this entry, login as root, cd into a uniVerse (or UniData) directory such as "fast". At the UNIX prompt, enter the command:

```
# crontab -l (the letter L in lowercase)
```

This will show the contents of the crontab directory for "root". You can overwrite this file or add to it. Normally we just add to the file. To do this we need a copy of the file. By redirecting the output from the "crontab" command we can accomplish this. The command looks like this:

```
# crontab -l > mytab
```

(make up a name which is NOT used in your current directory)

Now use an editor to edit "mytab" and add this command:

```
0 2 * * 0 (cd /u1/fast;/bin/sh run.setup) > /u1/fast/cron.out 2>&1
```

There are five date/time parameters that must be specified. These are (left to right) minute, hour, day, month, weekday. In my crontab the "0" is zero minutes, the "2" is 2am, the two asterisks are to fill the spaces for the day and month parameters; because we are going to run every Sunday, we don't want to specify a date (1 - 31) or a month (1 - 12). The "0" means run on Sunday. Monday is "1". (Please note that on some UNIX machines, Sunday is "7" instead of "0".)

On our HP the pathname to the crontabs directory is:

```
/usr/spool/cron/crontabs
```

I made a backup of the "root" entry before I proceeded. (This is a rule from the school of "hard knocks".) To make a backup I enter:

```
cp root root.save
```

"cp" is the UNIX copy command. A new file called "root.save" will be created which is identical to the root entry in this directory.

When I have completed changes, I used the UNIX command "crontab mytab" to register my changes with cron. The output from FAST will be in "cron.out" when FAST runs. To look at it you can use the UNIX commands:

cd /u1/fast

ls *(to see if it is there)*

cat cron.out | more *(or cat cron.out | pg)*

In working with cron I found the "log" file on our system to be in "/usr/lib/cron". I used "cat log|more" to look through the log to see that my cron job actually executed.

NOTE: All of the pathnames that we have used in this discussion are examples of MY system. The pathnames on your system will most likely be different. Be sure to find the correct paths to "uv" and to "fast" before you begin creating your scripts and crontab. If you are using Unidata, note that you must setup all of the Unidata variables (UDTHOME & UDTBIN). Obviously, instead of using "uv" you would use "udt".