NET INTEGRATION TECHNOLOGIES INC.

cron How-To

How to use cron to Schedule rsync Synchronizations

September 29, 2004

615-0006-01

cron How-To

In this How-to you will learn:

- What cron does
- How to install and configure a cron utility on a Nitix™ server
- How to configure Nitix to load cron as part of the boot sequence
- How to schedule a task to automatically run using cron
- How to schedule cron to run an rsync job

+++ NOTE +++

This How-To is provided as a service to our customers however, please be advised that you are working at your own risk as cron is NOT supported by our help desk personnel. Also this How-To assumes that you have a working knowledge of UNIX utilities, including cron and vi. Also, basic skills in working from a command line prompt from either a telnet session or a tty console session are assumed.

What cron does

Cron is a UNIX command for scheduling jobs to be executed sometime in the future. A cron is normally used to schedule a job that is executed periodically - for example, to send out a notice every morning. It is also a daemon process, meaning that it runs continuously, waiting for specific events to occur.

Installing the Nitix version of cron

Download a copy of the Nitix version of the cron utility from:

http://download.net-itech.com/public/cron.zip

Create a team called cron
NOTE: You are not required to assign members to, or create a password for, the team. If you do, you will use up one of your Nitix Client Access Licenses.

Copy the cron utility to the */home/cron/Files* directory

Make sure that cron is executable

chmod 765 /home/cron/Files/cron

Configuring cron to run every time Nitix boots

Create a directory called /bootdisk/etc

mkdir /bootdisk/etc

Create a file called ***rclocal*** in the */bootdisk/etc* directory with the following lines to get cron loaded during a reboot of the server.

#!/bin/sh

/home/cron/Files/cron

Do not include any spaces or blank lines before the "#!" symbols because they have to the very first characters in the file.

Reboot the server to load cron from rclocal

To confirm that cron is running:

From a telnet prompt or a console command prompt type:

ps aux | grep cron

The ps aux | grep cron command should return a line showing */home/cron/Files/cron* similar to the output shown below.

```
_root   5054  0.0  0.1  1744  588 ?   S  18:22  0:00 /home/cron/Files/cron
_root   4579  0.0  0.0  1464  468 ttyp0  R  18:33  0:00 grep cron
```

To load cron without rebooting the server; at a command prompt type:

/home/cron/Files/cron

Loading cron manually without rebooting the server does not confirm that your configuration is correct for loading cron automatically during a system reboot. Reboot the server at your earliest convenience and confirm that cron was loaded from the */bootdisk/etc/rclocal* file by typing the "ps aux | grep cron" command shown above.

Configuring cron

cron jobs are commands or script files that are scheduled to run at a predetermined time and frequency as defined in a user created cron table file.

To configure cron to schedule jobs automatically on a Nitix-powered server we first have to create a blank file called *crontab* in the */home/cron/Files* folder. The easiest way to do this is to use "touch," a UNIX command to create an empty file.

Enter the following command:

touch /home/cron/Files/crontab

(We won't actually be using the crontab file, but cron will look for it and complain if it isn't there.)

Next you have to create a new folder ***/home/cron/Files/crontabs*** to hold a file called *_root*. The *_root* file will hold a list of the jobs that will run with root privileges. To run a cron job as another user create a file by the users name in the crontabs folder.

To create the crontabs directory type:

mkdir /home/cron/Files/crontabs

Create the _root file by typing:

touch /home/cron/Files/crontabs/_root

Next change the permissions on the _root file by typing:

chmod 600 /home/cron/Files/crontabs/_root

This concludes the one time installation setup of cron. All that is left now is to define the jobs that you want to have cron to automatically run for you at the times you define in the _root file.

Configuring cron Jobs

You can enter any valid Nitix commands or script containing valid shell programming code.

cron jobs are defined on a per line basis in the *_root* file. You can enter any valid Nitix commands or scripts containing valid shell programming code as your command to execute.

Each line in the *_root* file defines a specific cron job and has six fields or parameters defined as follows:

```
1 2 3 4 5 6

   The time and date fields are:

      field            allowed values
      -----            --------------
1     minute           0-59
2     hour             0-23
3     day of month     1-31
4     month            1-12 (or names, see below)
5     day of week      0-7 (0 or 7 is Sun, or use names)
6     command to run any valid command or script


      The first five fields may be an asterisk (*), which
      stands for every minute, every hour, and so on.

      Ranges of numbers are allowed. Ranges are two numbers
      separated with a hyphen. The specified range is
      inclusive. For example, 8-11 for an ``hours'' entry
      specifies execution at hours 8, 9, 10 and 11.

      Lists are allowed. A list is a set of numbers (or ranges)
      separated by commas. Examples: ``1,2,5,9'', ``0-4,8-12''.

      The file may have any number of lines.

      Lines beginning with a # are comments, and are ignored.
```

For a more complete description of crontab syntax, review any good UNIX administration book or do an online search “using crontab.” Just remember that the crontab file we are working with here is called _root.

```
Example 1:

       # run five minutes after midnight, every day
       5 0 * * *       /home/cron/Files/bin/daily.job

Example 2:
       # run at 2:15pm on the first of every month
       15 14 1 * *     /home/cron/Files/bin/monthly.job
```

Remember your script files have to be executable and you should use absolute addressing to point to your script files as shown in the examples above.

Scheduling cron to run an rsync job

Now that we have cron installed and running we'll demonstrate how to create a *rsync.sh* script file to automatically run an rsync command on a recurring basis. In this example we'll configure the *rsync.sh* script to run the rsync command to synchronize files between folders on our two Nitix powered servers and then have cron execute the *rsync.sh* script at 5 minutes past midnight every Monday through Friday. See our ***rsync How-To*** for a more complete explanation of how rsync works.

The first step here is to get your rsync command tested and working from the command line before entering that command into the script file. In our example we'll synchronize the /home/artists/Files folder on a server called NITI1SRV to the /home/test/Files folder on the NITI2SRV server.

The command to do this by pushing the synchronization out from the NITI1SRV to the NITI2SRV would be:

rsync -zavP home/artists/Files/ cron@192.168.10.158::cron/Files*

Once you have this command working from the command line you can put the command into a file and then make that file executable. The name of the script file can be anything and in this example we'll call the file rsync.sh.

As you may use script files to automate many tasks it might be a good idea to create a bin folder below the cron home directory to hold all of our executable cron files. To create the bin directory enter:

mkdir /home/cron/bin

The next step is to create the script file, **rsync.sh,** with the following lines in the new **/home/cron/bin** folder.date

```
#!/bin/sh
RSYNC_PASSWORD=1234        #1234 is the password of the destination team account
export RSYNC_PASSWORD
rsync –zavP home/artists/Files/*  cron@192.168.10.158::cron/Files
```

We'll then make the file executable by entering:

chmod 770 /home/cron/bin/rsync.sh

Finally, you will need to enter your script file into the **_root** file to have it executed automatically by the cron service.

Open the /home/cron/Files/crontabs/_root file and enter the following command

```
#  run rsync.sh at 5 minutes past midnight Mon thru Fri
5 0 *  * 1-5      /home/cron/bin/rsync.sh
```

Save the file and you're done.

Summary

In this How-To we have learned how to install and configure the cron utility on a Nitix powered server. We then looked at configuring Nitix to load cron as part of the boot sequence and how to get cron to schedule a task to automatically run. We then learned how to schedule a cron job to synchronize folders between two Nitix powered servers using rsync on a recurring basis.

About Net Integration Technologies

Net Integration Technologies Inc. (NITI) is a software developer that delivers autonomic, Linux-based server operating system (OS) solutions to the SMB market. The company's revolutionary Nitix server OS sets new standards in stability, security, affordability and ease-of-use for small to mid-sized businesses. Nitix-powered servers have earned the company numerous awards and rave reviews from notable publications like PC Magazine, CRN, eWeek and InfoWorld. Established in 1997, the company has built a worldwide Approved Partnership Program with over 1,500 reselling partners. NITI is headquartered in Markham, Canada with additional offices in Montreal, the United States and Europe. For more information, please visit: http://www.nitix.com.