Job Scheduler
Utilities Manual

A component of Mark Dickinsons Unix Job Scheduler

This manual covers recomeded security settings for Marks Job Scheduler.
This manual is for version V1.13 of Marks Job Scheduler

Other reference material available

jobsched_cmd User guide

Job Scheduler Messages Manual

Job Scheduler Daemon Configuration Guide

Job Scheduler Sucity Guide

GPLV2 license is at http://www.gnu.org/licenses/gpl-2.0.html

Tested under Linux (Fedora, CentOS), and Solaris (2.8, 10 and OpenSolaris)

Table of Contents

1. Please Read This Section

The job scheduler application utilites are provided as standard, no issues there.

All the other utilities are provided as-is in the hope they may be usefull; they are not part of the job scheduler and are not maintained in conjunction with the job scheduler, they have their own CVS tree and seperate download pages on my website.

What you get shipped with the job scheduler in way of additional utilities is whatever I happened to have in place for testing at the time I build the job scheduler installation snapshot. If you want the latest versions you will need to download them from my website.

This manual will probably be obsoleted in the next release of the job scheduler. It is intended that as the jobsched_take_snapshot is fully documented in the scheduler daemon configuration guide that the jobutil utility will also be documented there. All the non-scheduler utility docuentation will be removed from distribution; it simply doesn't have anything to do with the job scheduler.

2. Utilites that are part of the Job Scheduler application

2.1 jobsched_take_snapshot – backup/restore databases

The jobsched_take_snapshot program is used to backup the scheduler databases into a text file containing the commands needed to completely recreate the job scheduler databases to the point in time they were at at the time the snapshot was taken.

How to use this utility is covered fully in the Job Scheder Daemon configuration guide, as it should be run (the backups at least) as part of a daily scheduler maintenance task.

But basically for backup

cd /scheduler/install/path (ie: cd /opt/dickonson/job_scheduler, wherever you put it)
bin/jobsched_take_snapshot ./ > any_backup_filename.txt

The only parameter it accepts is the path to the scheduler directory, which must be provided.

The example used above uses ./ as I always cd to the directory simply because jobsched_take_snapshot is not, and does not need to be, in a program search path.

For restore... refer to the job scheduler daemon cofiguration guide, only system administrators should be restoring back to prior states.

2.2 Jobutil – job database display utility

The jobutil utility is a work in progress; stalled progress.

It was primarily a quck way of looking at the job scheduler job database as a debugging tool. As the job scheduler does not display jobs in execution order but in the order they were added I wanted an execution order list, and as the scheduler doesn't display the internal job database I wanted a way to display deleted job records to ensure they really did get totally removed on the scheduler newday databaee compress activity.

At one point I indented it to be a do-all end-all utility until I decided I really didn't need it at all. Adding further functionality to this has pretty much stopped.

The only other thing I might have done with it was use it to sort the entries in the job database so they would be displayed sorted when scheduler commands are used, but I never did decide if I wanted them sorted by jobname or execution time order so never implemented that.

Syntax: jobutil -d */path/to/install/database/dir* **-c** *command*

command may be one of

- execreport – show jobs in the job database by sorted by their next execution time order, excluding any deleted jobs
- namereport – show jobs in the job database sorted by jobname, excluding any deleted jobs
- deletelist – show jobs that have been flagged as deleted, but still in the database as the

scheduler newday has not run yet

- sortdbs – **never implemented**

Examples

```
[mark@falcon job_scheduler]$ ./jobutil -d ./ -c execreport
Jobs in execution order
 1. 20040619 16:50:00 SYS-R030-FILESIZE-CHECK
 2. 20040619 16:55:00 SYS-R015-SYSTEM-CHECK
 3. 20040619 17:00:00 SYS-R060-SYSTEMSTATS
 4. 20040619 17:30:00 JS_-DAILY-SNAPSHOT
 5. 20040619 17:30:00 SYS-DAILY-DISK-CLEANUP
 6. 20040619 18:00:00 SEC-DAILY-SCHEDAUTH-CHECKS
 7. 20040619 22:00:00 SYS-RPM-BACKUP
 8. 20040619 22:15:00 SYS-DAILY-SOURCE-BACKUP
 9. 20040619 22:30:00 SEC-DAILY-FPROT-VIRUS-SCAN
10. 20040619 23:00:00 JS_-DAILY-LOG-CLEANUPS
11. 20040619 23:30:00 SEC-DAILY-LOG-ARCHIVE
12. 20040619 23:45:00 SEC-DAILY-ARCHIVE-CLEANUP
13. 20040620 00:01:00 WEB-DAILY-ACCESS-STATS
14. 20040620 01:00:00 SYS-ROLL-MESSAGES-FILE
15. 20040620 04:30:00 SYS-DAILY-ALERT-RPT
16. 20040620 07:15:00 JS_-DAILY-NEWDAYCHK
17. 20040620 12:00:00 DAILY-DISK-USAGE-HIST
18. 20040620 21:30:00 NULL-ARCHIVE-JOBLOGS
19. 20040625 23:30:00 SEC-WEEKLY-TRIPWIRE

19 jobs defined

[mark@falcon job_scheduler]$ ./jobutil -d ./ -c namereport
 Jobs in name order
 1. DAILY-DISK-USAGE-HIST
 2. JS_-DAILY-LOG-CLEANUPS
 3. JS_-DAILY-NEWDAYCHK
 4. JS_-DAILY-SNAPSHOT
 5. NULL-ARCHIVE-JOBLOGS
 6. SEC-DAILY-ARCHIVE-CLEANUP
 7. SEC-DAILY-FPROT-VIRUS-SCAN
 8. SEC-DAILY-LOG-ARCHIVE
 9. SEC-DAILY-SCHEDAUTH-CHECKS
10. SEC-WEEKLY-TRIPWIRE
11. SYS-DAILY-ALERT-RPT
12. SYS-DAILY-DISK-CLEANUP
13. SYS-DAILY-SOURCE-BACKUP
14. SYS-R015-SYSTEM-CHECK
15. SYS-R030-FILESIZE-CHECK
16. SYS-R060-SYSTEMSTATS
17. SYS-ROLL-MESSAGES-FILE
18. SYS-RPM-BACKUP
19. WEB-DAILY-ACCESS-STATS

19 jobs defined

[mark@falcon job_scheduler]$ ./jobutil -d ./ -c deletelist
Recently deleted jobs...
SYS-R005-PROCBUSY-CHECK
SYS-R005-LOADAVG-CHECK
```

```
SYS-R010-CHECK-FREEMEM
SYS-R009-CHECK-DISK-BUSY
SEC-DAILY-TRIPWIRE
SEC-DAILY-LOG-ARCHIVE
TEST-FAIL
TEST-FAIL-CLEAR
SEC-WEEKLY-TRIPWIRE
SEC-WEEKLY-TRIPWIRE

There are 10 entries that should be removed in the next newday
```

3. Utilities that are not part of the Job Scheduler application

3.1 So why these utilities then ?

I found issues with portability in my scripts when moving them between Linux/Solaris/AIX so these utilities were created in order that I could compile them on the target server the scripts would run on; and therefore my job batch scripts could then be used unmodified on each platform.

They are nothing fancy, they are just intended to provide scripts with exactly the information they want without me having to use uname and lots of case statements for every possible OS in my scripts.

You will find them in the bin/my_tools directory under the directory you installed the job scheduler into.

As the job scheduler is currently shipped these are all Linux ELF 32 bit executables in the bin/my_tools directory, but the source for these utilities should also be in the *source* directory under the instalation path so you can use GCC to build copies for non Linux OS's.

3.2 daysago utility

All this utility does is tell you how many days in the past a date was.

If provided a date in YYYYMMDD format will return the number of days in the past the date ocurred on. If the date is in the future will return 0.

Note: **If the date is invalid will return an error string rather than a number.**

Syntax: **daysago YYYYMMDD**

```
Example:
   [mark@myhost temp]$ date
   Sun Jul 28 09:23:38 EDT 2002
   [mark@myhost temp]$ ./daysago 20020726
   2
```

3.3 workoutdate utility

This utility is used to workout the date string of a date n number of days either side of the current date and return it in a variety of formats. If you do not wish the current date to be used as a base for offsets you may override the date to be used.

This will calculate offsets with a mixture of days and hours.

History: I got sick of trying to do this in scripts (although I did eventually create a script library that does this and a lot more the script library is much slower).

```
Syntax:
[mark@myhost temp]$ ./workoutdate
      ./workoutdate: (c)Mark Dickinson, 2001

      Parameters available are
        -daysback nn
        -daysforward nn
        -hoursback nn
        -hoursforward nn
        -format YYYYMMDD | YYMMDD | YYYYMMDDHHMM | HHMM | CTIME
        -workingdate YYYYMMDDHHMM  (only use if you don't want
         calculations from the current time, see important nore below)

      If a -format is not provided the default is YYYYMMDD
      If a -workingdate is not provided the current date and
        time will be usd.
      The -format CTIME displays the time as per the ctime function
        on your machine

      Parameters may be mixed, ie: -daysback 5 -daysforward 2

Examples:
  [mark@myhost temp]$ date
  Sun Jul 28 09:49:08 EDT 2002

  [mark@myhost temp]$ ./workoutdate -daysback 2 -format YYYYMMDDHHMM
  200207260949

  [mark@myhost temp]$ ./workoutdate -daysback 2 -format CTIME
  Fri Jul 26 09:49:36 2002

  [mark@myhost temp]$ ./workoutdate -workingdate 200207260949 \
         -daysback 2 -format CTIME
  Wed Jul 24 09:49:00 2002
```

Important note: sanity checking not implemented in the -workingdate option, a trap if you forget the HHMM as below.

```
   [mark@myhost temp]$ ./workoutdate -workingdate 20020726 \
         -daysback 2 -format CTIME
   Wed Jul 24 00:00:00 2002
```

3.4 datelib_ksh – shell script date library

This is a ***ksh shell*** script library that can do absolutely anything you are ever likely to want to do with a date.

It is obviously a lot slower than using one of the compiled date utility programs above.

Also as noted this is in the section on utilies not part of the job scheduler;
this ksh shell script library is maintained on a totally seperate CVS tree and is a totally seperate download from my website; the version shipped with the job scheduler is unlikely to be the latest version and will never have any bug fixes applied. While I do occasionally update the install package for the job scheduler with a newer version it is infrequent.

Minimal documentation on function calls from the date library are contained in comments at the end of the file. The latest documentation on the ksh library at the time of this document being created is at
http://mdickinson.dnsalias.org/linux/manuals/date_tools/shell_date_library/datelib_doc.html

3.5 filesize

This utility was created as a support utility for some of my scripts; where I wanted to check error log files (if empty, no errors :-)) from scripts,

I use it to check error logs, if an error log from a job does not have an end-of-file the log can be emailed to me or an alert raised. It's a simple utility I can use to stop myself getting undeeded emails from empty log files.

It simply reports the filesize of a file as it's output. If the file is empty it will report 0.

Believe it or not this is better that trying to script filesize checks, 'ls -la' output has changed format has changed between fedora versions alone (lots of script breaks there) never mind complications with differences on other OS's; this program will always simply return the filesize.

Example:
[mark@falcon my_tools]$ ./filesize workoutdate
9633
[mark@falcon my_tools]$ ls -la workoutdate
-rwxr-xr-x. 1 mark mark 9633 Mar 13 13:05 workoutdate