



# Shepherd.Cron User's Guide

*J&J Computer Consulting Shepherd Server Publications*

## Overview

Shepherd.Cron provides rudimentary job scheduling services from within Shepherd. Implemented as an Agent, Shepherd.Cron supports full remote management taking all job information from Shepherd Directory Services (SDS) and responding to load and unload requests as required.

Shepherd.Cron can run functions exported from DLLs and/or Shepherd commands. At this time, starting external programs is not supported by Shepherd.Cron, but it could be implemented through the DLL interface if necessary.

Job scheduling, much like Unix, can consist of any combination of the following parameters:

- ◆ Minute
- ◆ Hour
- ◆ Day of the Month
- ◆ Month of the Year
- ◆ Day of the Week

A wild card (\*) can stand in for any of these values to support various combinations of job schedules.

## Configuring the Shepherd.Cron Agent

In most Shepherd installations, the Shepherd.Cron Agent files can already be found in the agents subdirectory. If not, copy cron.dll and cron.agt from a Shepherd.Cron distribution into the agents subdirectory before continuing.

At this point, setting up the Shepherd.Cron Agent is no different than setting up any other Shepherd Agent. An example ShepherdAgent object for Shepherd.Cron should look like:

```
dn: cn=cron, c=US
cn: cron
objectclass: ShepherdAgent
agentModule: agents\cron.agt
hostServer: cn=Server1, c=US
log: cn=cronLog, c=US
```

The agentModule directive points to the cron.agt file, and the hostServer should point to the distinguished name of the Shepherd Server on which Shepherd.Cron should run. Multiple hostServer values can be used if the configuration matches across multiple servers.

Shepherd.Cron supports only one log file, so only one value should be listed. Additionally, a logType attribute is not needed in the referenced object. Typically, Agent logs are referred to with an objectclass of "Log" as opposed to the "ServiceLog" objectclass traditionally used for Shepherd Services. The "ServiceLog" objectclass remains only for backward compatibility with previous releases. The Log object needs only a filename attribute in addition to the objectclass to be complete.

Once the ShepherdAgent and Log objects are configured, you should be ready to load the Shepherd.Cron Agent. You can accomplish this by restarting Shepherd or by loading the Shepherd.Cron Agent through the console or Shepherd.Admin.

## Scheduling Jobs

Shepherd.Cron defines an additional objectclass and several attributes for scheduling jobs. The objectclass "Job" defines an object within SDS as a Shepherd.Cron Job. Shepherd.Cron uses the following attributes to schedule the job:

Attribute	Possible Values*	Description
minute	0-59	Specifies the minute of the hour at which the job will run.
hour	0-23	Specifies the hour of the day at which the job will run.
dayOfMonth	1-31	Specifies the day of the month on which the job will run.
monthOfYear	1-12	Specifies the month of the year on which the job will run.
dayOfWeek	0-7	Specifies the day of the week on which the job will run. 0 and 7 both reference Sunday.

\* The wildcard character (\*) can be used to match all possible values.

The only required attribute for a Job object is minute. Shepherd.Cron assumes that any other attribute not specified should match all possible values as if a wildcard had been specified for the attribute. To prevent a heavy load due to a misconfiguration, Shepherd requires minute to keep from running jobs once per minute that should not be run every minute.

Shepherd.Cron uses the hostServer attribute to determine what servers should run the Job. The hostServer attribute can be multi-valued if the Job needs to run on more than one Shepherd server.

Shepherd.Cron requires one of two additional attributes for defining the job. The **command** attribute identifies the Shepherd Console command that Shepherd.Cron should run for the job. The system accepts multiple command values for a single Job. The **api** attribute defines a function that Shepherd.Cron should call within an external DLL. The api attribute format follows:

<dll name>!<method name>

Shepherd accepts relative or absolute <dll name> values. Shepherd.Cron logs an error if it is unable to load the DLL and/or access the requested method. At this time, no parameters are passed to external api's from Shepherd.Cron.

The following LDIF data defines a simple job for rotating the cron log files daily at midnight.

```
dn: cn=rotateLogs, c=US
cn: rotateLogs
objectclass: Job
command: log rotate cn=cronLog, c=US
minute: 0
hour: 0
```

This entry specifies that the command "log rotate cn=cronLog, c=US" should run at 00:00 (midnight) every day. The scheduling attributes not provided are assumed to be wildcards causing the Job to run daily.

Be sure to restart the Shepherd.Cron Agent when new jobs are added or old when old jobs are deleted. Since Shepherd.Cron loads the jobs during initialization for efficiency, it cannot find out about new jobs unless it is re-initialized.