Cron Service User's Guide

Version 1.2.6

Last updated: March 29, 2006

Overview

The Cron service allows you to register STAF commands that will be executed at a specified time interval(s).

Note that Cron registration information is persistent data. This means that if you register with the Cron Service, if you shutdown STAF and restart it (even if you reboot the machine), the prior registration information will still be active. When STAF starts, it reads in the previous Cron registration information, and will execute the registered STAF commands at the specified time interval(s).

Note that the Cron service uses its machine's Operating System date/time information to determine the current date/time in relation to the registered requests. Users of the Cron service must ensure that the machine on which the Cron service is running has the correct Operating System date/time (and that it correctly updates the current date/time relative to Daylight Savings Time).

Installation and Configuration

Install Java 1.2 or later.

Install STAF Version 2.6.0 or later (but less than Version 3.0) by following the installation instructions in the STAF documentation.

Download the CronV126.zip/tar file from Download Services for STAF V2 into a local directory (e.g. C:/STAF/services or /usr/local/staf/services) and extract it.

Configure the Cron service:

Add the following statement to your staf.cfg file:

```
service <CronName> LIBRARY JSTAF EXECUTE c:/staf/services/STAFCron.jar
```

where:

- `<CronName>` is the name by which the Cron service will be known on this machine.

Examples:

```
service cron LIBRARY JSTAF EXECUTE c:/staf/services/STAFCron.jar

service cron LIBRARY JSTAF EXECUTE /usr/local/staf/services/STAFCron.jar
```

Request Syntax

The STAX service provides the following requests:

- REGISTER - Registers a STAF command to be executed at a specified time interval
- UNREGISTER - Unregisters a STAF command
- LIST - Lists registered STAF commands
- HELP - Displays a list of requests for the Cron service and how to use them.

REGISTER

REGISTER registers a STAF command to be executed at a specified time interval

Syntax

```
REGISTER   MACHINE <machine> | PYTHONMACHINE <machine>
           SERVICE <service> | PYTHONSERVICE <service>
           REQUEST <request> | PYTHONREQUEST <request>
           [PREPARE <script>]
           [MINUTE <minute>] [HOUR <hour>]
           [DAY <day>] [MONTH <month>]
           [WEEKDAY <weekday>]
```

`MACHINE` specifies the name of the machine where the command will be executed. This option will resolve STAF variables. When the specified time interval occurs, the value of `MACHINE` will not be evaluated as a python string.

`PYTHONMACHINE` specifies the name of the machine where the command will be executed. This option will not resolve STAF variables. When the specified time interval occurs, the value of `MACHINE` will be evaluated as a python string.

`SERVICE` specifies the name of the service to be executed. This option will resolve STAF variables. When the specified time interval occurs, the value of `SERVICE` will not be evaluated as a python string.

`PYTHONSERVICE` specifies the name of the service to be executed. This option will not resolve STAF variables. When the specified time interval occurs, the value of `SERVICE` will be evaluated as a python string.

`REQUEST` specifies the request to be executed. This option will resolve STAF variables. When the specified time interval occurs, the value of `REQUEST` will not be evaluated as a python string.

`PYTHONREQUEST` specifies the request to be executed. This option will not resolve STAF variables. When the specified time interval occurs, the value of `RQUEST` will be evaluated as a python string.

`PREPARE` specifies Python code which will be executed when the specified time interval occurs. This code will be executed prior to the `PYTHONMACHINE`, `PYTHONSERVICE`, and `PYTHONREQUEST` options being evaluated as python strings. This option will not resolve STAF variables. If the Python code sets the variable
STAFCronSubmit to any string other than 'true', then the request will not be submitted.

`MINUTE` specifies the exact minute(s) that the request executes. The valid values are 0 - 59. You can specify either * (an asterisk) or ANY, meaning all valid values, or a list of elements separated by commas. An element is either a number or an inclusive range, indicated by two numbers separated by a minus sign (such as 1-15).

`HOUR` specifies the exact hour(s) that the request executes. The valid values are 0 - 23. You can specify either * (an asterisk) or ANY, meaning all valid values, or a list of elements separated by commas. An element is either a number or an inclusive range, indicated by two numbers separated by a minus sign (such as 8-17).

`DAY` specifies the exact day(s) that the request executes. The valid values are 1 - 31. You can specify either * (an asterisk) or ANY, meaning all valid values, or a list of elements separated by commas. An element is either a number or an inclusive range, indicated by two numbers

separated by a minus sign (such as 1-2).

MONTH specifies the exact month(s) that the request executes. The valid values are 1 - 12. You can specify either * (an asterisk) or ANY, meaning all valid values, or a list of elements separated by commas. An element is either a number or an inclusive range, indicated by two numbers separated by a minus sign (such as 5-6).

WEEKDAY specifies the exact weekday(s) that the request executes. The valid values are 0 - 6 (Sunday = 0, ...). You can also specify the textual weekday ("Sunday", etc.). You can specify either * (an asterisk) or ANY, meaning all valid values, or a list of elements separated by commas. An element is either a number or an inclusive range, indicated by two numbers separated by a minus sign (such as 1-15).

Note: At least one of the time interval options (MINUTE, HOUR, DAY, MONTH, WEEKDAY) must be specified in a REGISTER request.

Security

This request requires at least trust level 5.

Results

Upon successful return, the result buffer contains the Cron ID.

Examples

Goal: Execute the command "start command notepad" at 1:00AM every day.

```
cron register machine local service process request "start command notepad" hour 1
```

Goal: Execute the command "start command notepad" at 1:00AM every Sunday.

```
cron register machine local service process request "start command notepad" hour 1 weekday Sunday
```

Goal: Execute a STAX job every December 1st.

```
cron register machine local service stax request "execute file c:/tests/startall.xml" month 12
```

Note that the registration in this example would only result in the command being executed the next time the month changes to 12. So, if you registered this command on Dec 5, 2003, it would not be executed until 12AM Dec 1 2004.

Goal: Execute a STAX job every day in December.

```
cron register machine local service stax request "execute file c:/tests/startall.xml" month 12 day *
```

Goal: Execute a STAX job called C:/automate/updateStatus.xml via the STAX service on machine server1 every week day (Monday through Friday) at the following times: 8:00AM, 12:00PM, 3:00PM, 4:00PM, 5:00PM, and midnight.

```
STAF local CRON REGISTER MACHINE server1 SERVICE stax REQUEST "EXECUTE FILE
c:/automate/updateStatus.xml" WEEKDAY "Monday-Friday" HOUR "8, 12, 15-18, 0"
```

UNREGISTER

UNREGISTER unregisters a STAF command with the Cron Service.

Syntax

```
UNREGISTER ID <registrationID>
```

`ID` specifies the Cron ID which is to be unregistered.

Security

This request requires at least trust level 3.

LIST

LIST lists information about the commands registered with the Cron service.

Syntax

```
LIST [MACHINE <machine>]
```

`MACHINE` specifies the machine for which Cron requests should be listed.

Results

Upon successful return, this request will return the matching information for any registered Cron requests,
in the format:
cronID;machineName;machineScriptFlag;serviceName;serviceScriptFlag;request;requestScriptFlag;prepare;minute;hour;day;month;weekday;

```
2;local;false;process;false;start command notepad;false;;;;ANY;;;;
1;local;false;process;false;start command notepad;false;;1;;;;;
```

VERSION

VERSION displays the CronService version.

Syntax

```
VERSION
```

Results

The result is the version number of the Cron service.

HELP

HELP displays the request options and how to use them.

Syntax

```
HELP
```

Results

The result buffer contains the Help messages for the request options for the Cron service.

Cron Service Logging

The Cron service maintains a machine log where it writes an entry when the following occurs:

- A REGISTER request is received
- A time trigger matches the options previously specified in a REGISTER request. The log entry will include the STAF request number for the submitted STAF command.
- The service receives notification that a submitted STAF command has completed. The log entry will include the STAF request number, as well as the RC and result from the STAF command.
- An UNREGISTER request is received

The logname for the Cron service is the name under which the service is registered.

Here is an example of what the Cron service log looks like:

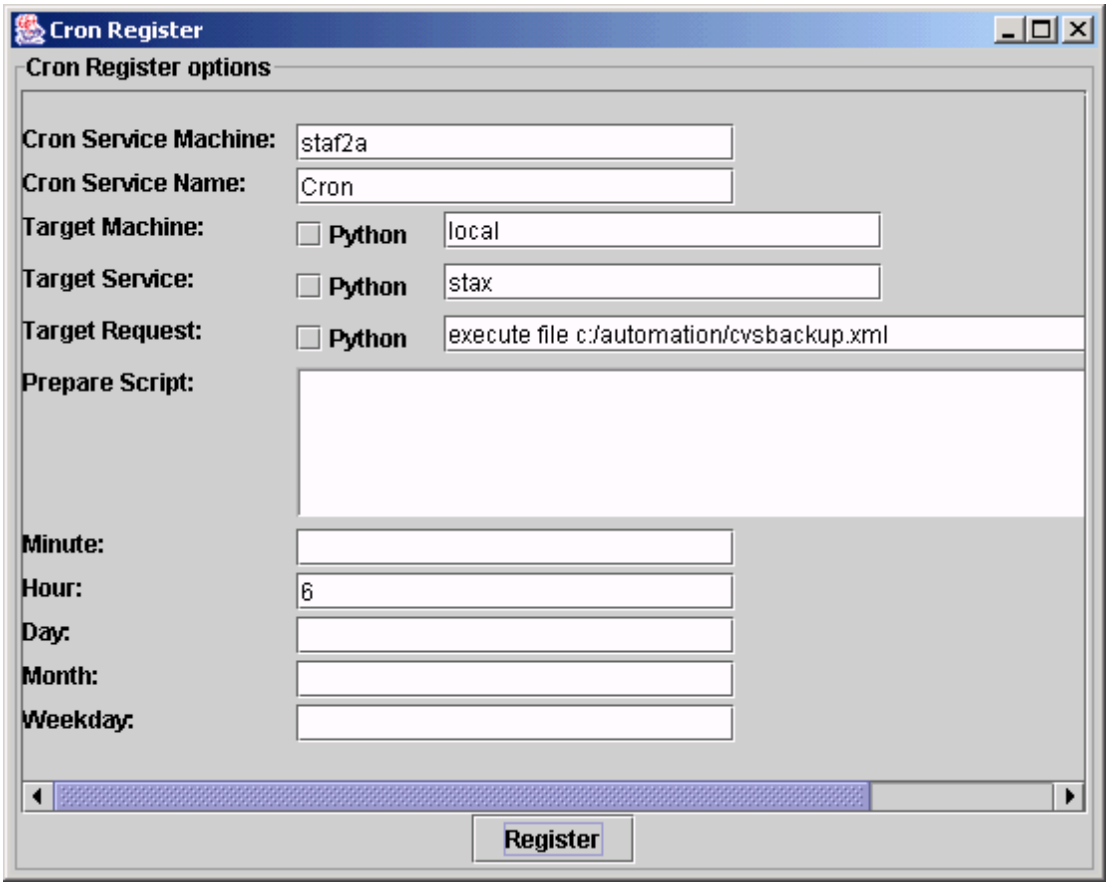
```
20040131-00:05:45|78ntmm3|8|STAF/SERVICE/cron|Info|Register ID=1
originMachine=78ntmm3.austin.ibm.com machine=78ntmm3 pythonMachine=false service=stax
pythonService=false request=version pythonRequest=false prepare= minute=[ANY] hour=[]
day=[31] month=[] weekday=[]
20040131-00:06:00|78ntmm3|8|STAF/SERVICE/cron|Info|Register ID=2
originMachine=78ntmm3.austin.ibm.com machine=78ntmm3 pythonMachine=false
service=event pythonService=false request=blah pythonRequest=false prepare=
minute=[ANY] hour=[] day=[31] month=[1] weekday=[]
20040131-00:06:08|78ntmm3|8|STAF/SERVICE/cron|Info|Submitted request #120 ID=2
triggers=minute[6],day[31],month[1] machine=78ntmm3 service=event request=blah
20040131-00:06:08|78ntmm3|8|STAF/SERVICE/cron|Info|Submitted request #124 ID=1
triggers=minute[6],day[31] machine=78ntmm3 service=stax request=version
20040131-00:06:08|78ntmm3|8|STAF/SERVICE/cron|Error|Request #120: RC=7, Result=blah,
Request submitted: machine=78ntmm3 service=event request=blah
20040131-00:06:08|78ntmm3|8|STAF/SERVICE/cron|Info|Request #124: RC=0, Result=1.5.0,
Request submitted: machine=78ntmm3 service=stax request=version
20040131-00:07:08|78ntmm3|8|STAF/SERVICE/cron|Info|Submitted request #142 ID=2
triggers=minute[7],day[31],month[1] machine=78ntmm3 service=event request=blah
20040131-00:07:08|78ntmm3|8|STAF/SERVICE/cron|Info|Submitted request #146 ID=1
triggers=minute[7],day[31] machine=78ntmm3 service=stax request=version
20040131-00:07:08|78ntmm3|8|STAF/SERVICE/cron|Error|Request #142: RC=7, Result=blah,
Request submitted: machine=78ntmm3 service=event request=blah
20040131-00:07:08|78ntmm3|8|STAF/SERVICE/cron|Info|Request #146: RC=0, Result=1.5.0,
Request submitted: machine=78ntmm3 service=stax request=version
20040131-00:07:44|78ntmm3|8|STAF/SERVICE/cron|Info|Unregister ID=1
20040131-00:07:45|78ntmm3|8|STAF/SERVICE/cron|Info|Unregister ID=2
```

Cron Registration GUI

The Cron Service provides a GUI to simplify the Cron Registration. To use this GUI, after installing and configuring the Cron Service, from a command prompt enter the following command (note that to run this command, the STAFCron.jar file must be in your CLASSPATH):

```
java com.ibm.staf.service.cron.CronRegister
```

A GUI will be displayed which allows you to enter the Cron Registration options.



The "Cron Service Machine" default is the local machine. The "Cron Service Name" default is "Cron".

Fill in the required values for "Target Machine", "Target Service", and "Target Request". Optionally, you may specify Python code in the "Prepare Script" field.

Fill in one or more values for "Minute", "Hour", "Day", "Month", and "Weekday".

This example shows the options you would specify to execute a STAX job at 6:00am every day.

Appendix A: Licenses and Acknowledgements

Jython

Jython is an implementation of the high-level, dynamic, object-oriented language Python written in 100% Pure Java, and seamlessly integrated with the Java platform. It thus allows you to run Python on any Java platform.

Acknowledgement

This product includes software developed by the Jython Developers (http://www.jython.org/).

Licence

```
Jython Software License
=======================
```