



Cron Converter

Job Scheduler

Dokumentation

Februar 2008

Impressum

Software- und Organisations-Service GmbH
Giesebrechtstr. 15

D-10629 Berlin

Telefon (030) 86 47 90-0

Telefax (030) 8 61 33 35

Mail info@sos-berlin.com

Web <http://www.sos-berlin.com>

Letzte Aktualisierung: Februar 2008

Inhaltsverzeichnis

1 Einführung	4
1.1 Voraussetzungen	5
2 Das Cron Converter Kommandozeilen Tool	6
3 Der Cron Adapter Job	8
3.1 Der Cron Adapter Job mit zentraler Konfiguration	9
4 User crontab oder System crontab?	11
Anhang A: Parameter	12
A.1 Steuern der Job-Konvertierung	12
Anhang B: Einschränkungen	13

1 Einführung

Wozu einen Job Scheduler einsetzen, wenn Cron als Scheduler im Betriebssystem enthalten ist?

Für die Verwendung von Cron spricht die Einfachheit der Konfiguration und die Kopplung mit Unix-Kommandos. Einige Gründe für einen besseren Job Scheduler als Cron:

- Cron führt keine Historie
Cron startet Jobs, speichert aber keine Informationen darüber, wann ein Job gestartet und beendet wurde und mit welchem Ergebnis. Sie können mit Cron keine auskunftsfähige bzw. revisionsfähige Job-Historie bereitstellen.
Der Open Source Job Scheduler hält die Job-Historie in einer Datenbank bereit und bietet ein Web Interface für die Historienauskunft.
- Cron führt kein Protokoll
Die Ausgaben Ihrer Jobs (stdout/stderr) werden von Cron ignoriert. Sie müssen daher in Ihren Jobs selbst die Handhabung von Protokolldateien implementieren und diese nachträglich zentralisiert einsammeln.
Der Open Source Job Scheduler sammelt automatisch die Ausgaben nach stdout/stderr der Jobs ein und speichert die Protokolle zentral bzw. wahlweise komprimiert in der Datenbank.
- Cron arbeitet nach dem Prinzip fire & forget
Es werden wenig Möglichkeiten angeboten, auf das Verarbeitungsergebnis eines Jobs zu reagieren. Sie können zwar unter Unix die Verarbeitungskontrolle selbst implementieren, werden hierbei aber erheblichen Aufwand haben und das Rad neu erfinden müssen.
Der Open Source Job Scheduler kann im Fall fehlerhafter Job-Läufe Benachrichtigungen per E-Mail versenden und Folge-Jobs zur Fehlerbehandlung starten. Darüber hinaus bietet er ein Web-Interface, in dem die nächsten Job-Startzeiten sichtbar sind und Jobs kontrolliert werden können, z.B. ad hoc starten, stoppen, pausieren.
- Cron kennt keine Job-Ketten
Die Konkatenierung mehrerer Kommandos zur Ausführung in Cron stellt nicht bereits eine Job-Kette dar. Job-Ketten kennen Abhängigkeiten der Job-Schritte voneinander und können je nach Verarbeitungsergebnis andere Folge-Jobs in einer Kette starten.
Der Open Source Job Scheduler kennt Job-Ketten mit Abhängigkeiten und hält deren Zustände persistent in einer Datenbank, um nach einem Systemausfall automatisch wiederanlaufen zu können.

Der *Cron Converter* des Open Source Job Schedulers stellt einen Migrationspfad für Cron dar, bei dem

- die Konfiguration mittels `crontab` erhalten bleiben kann
Der Job Scheduler liest automatisch eine vorhandene `crontab` ein und erzeugt daraus Job-Definitionen. Das Einlesen erfolgt analog zu Cron alle 60 Sekunden.
Diese Jobs nutzen bei weitem nicht alle Fähigkeiten des Job Schedulers, werden aber in einer Historie geführt, erhalten automatisch zentrale Protokolle und können mit einem Web-Interface kontrolliert werden.
- oder die `crontab` in das Format der Job Scheduler Konfigurationsdateien transformiert wird
In diesem Fall verwenden Sie `crontab` als Ausgangspunkt Ihrer Migration und konfigurieren die migrierten Jobs weiter mit dem graphischen Editoren des Job Schedulers.

Der Job Scheduler stellt zwei einfache Mechanismen zur Verfügung, um einen Umstieg von Cron auf den Job Scheduler zu vereinfachen:

- Das *Cron Converter* Kommandozeilen Tool
- Der *Cron Adapter Job*

Die Unterschiede sind in den folgenden Kapiteln dargestellt.

1.1 Voraussetzungen

Der Cron Adapter Job und Das Cron Converter Kommandozeilen Tool befinden sich im sos.scheduler.jar, das mit dem Job Scheduler ausgeliefert wird.

Zur Verwendung wird ein Java Runtime Environment ab Version 1.4 benötigt.

2 Das Cron Converter Kommandozeilen Tool

Das *Cron Converter* Kommandozeilen Tool ist für eine einmalige Migration einer `crontab` Datei zu einer Job Scheduler XML Konfiguration gedacht. Es analysiert eine `crontab` Datei und erstellt daraus eine XML-Datei, die per `<base file = "my_cron_configuration.xml"/>`

in die Job Scheduler Konfigurationsdatei `./config/scheduler.xml` eingebunden werden kann.

Das Kommandozeilen Tool wird mit dem Script `cronconverter.sh` gestartet, das sich im `./bin` Verzeichnis der Job Scheduler Installation befindet.

Parametrisierung siehe Anhang (Seite 12)

Beispiel: Verwendung des Kommandozeilen Tools

Die Datei `/home/scheduler/scheduler/crontab` hat den folgenden Inhalt:

```
5 * * * * /usr/bin/message.sh
00 */2 * * * /usr/local/bin/mail_poll
59 23 * * 0 cp /var/log/messages /log/backup/messages
```

Aus dem Verzeichnis des Job Schedulers (hier `/home/scheduler/scheduler`) kann die crontab mit dem folgenden Aufruf zu einer XML Konfigurationsdatei konvertiert werden:

```
bin/cronconverter.sh -crontab crontab -target config/my_scheduler_cron.xml
```

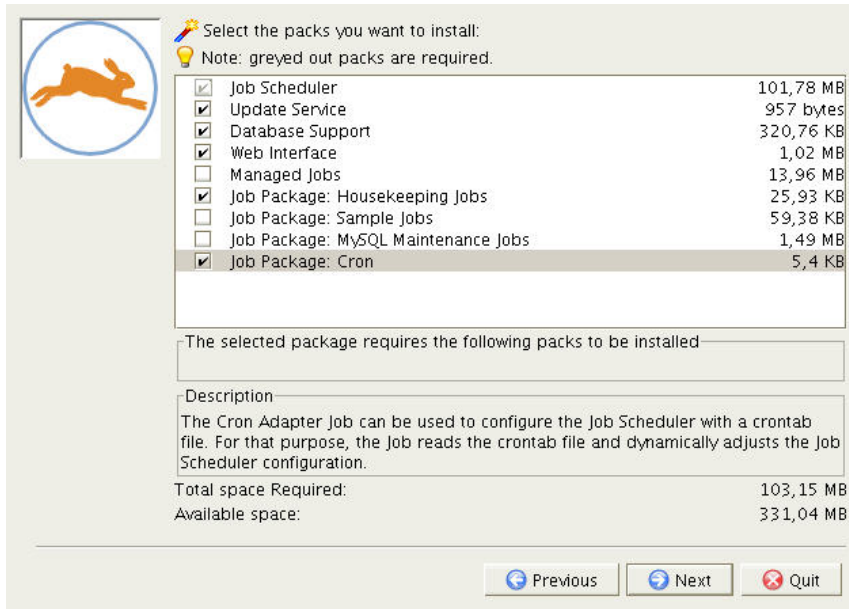
Der Inhalt der Konfigurationsdatei `config/my_scheduler_cron.xml` ist dann:

```
<?xml version="1.0" encoding="UTF-8"?>
<spooler>
  <config>
    <jobs>
      <job name="usr_bin_message.sh" timeout="600"
        title="Cron Job /usr/bin/message.sh">
        <script language="shell">/usr/bin/message.sh</script>
        <run_time>
          <period absolute_repeat="01:00" begin="00:05"/>
        </run_time>
      </job>
      <job name="usr_local_bin_mail_poll" timeout="600"
        title="Cron Job /usr/local/bin/mail_poll">
        <script language="shell">/usr/local/bin/mail_poll</script>
        <run_time>
          <period absolute_repeat="02:00" begin="00:00"/>
        </run_time>
      </job>
      <job name="var_log_messages" timeout="600"
        title="Cron Job cp /var/log/messages /log/backup/messages">
        <script language="shell">
          cp /var/log/messages /log/backup/messages
        </script>
        <run_time>
          <weekdays>
            <day day="0">
              <period single_start="23:59"/>
            </day>
          </weekdays>
        </run_time>
      </job>
    </jobs>
  </config>
</spooler>
```

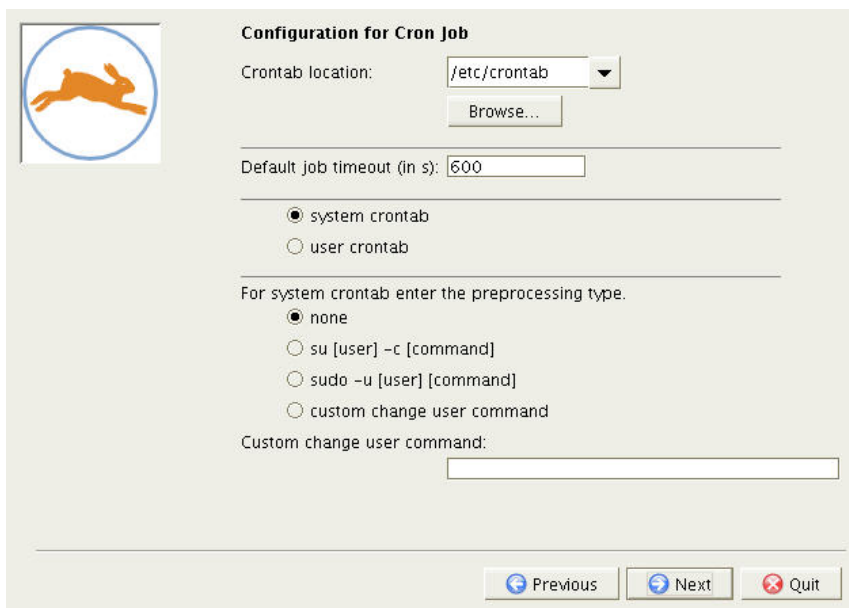
3 Der Cron Adapter Job

Der *Cron Adapter Job* kann genutzt werden, um den Job Scheduler über eine `crontab` Datei zu konfigurieren. Dazu liest der Job die `crontab` Datei und passt die Konfiguration des Job Schedulers zur Laufzeit an.

Um den *Cron Adapter Job* zu verwenden, muss das Cron-Paket im Setup angewählt werden.



Das Setup fragt die für den Job benötigten Parameter (Lage der `crontab`, Art der `crontab`, ...) ab und richtet den Job in der Konfigurationsdatei `./config/scheduler_cron.xml` ein.



Der Job wird dann vom Setup wie folgt konfiguriert:

```
<job name      = "scheduler_cron_adapter"
  title        = "Read crontab"
  spooler_id   = "">
```

```
<description>
  <include file = "jobs/JobSchedulerCronAdapter.xml"/>
</description>

<params>
  <param name="crontab" value="crontab"/>
  <!-- 0 for User crontab, 1 for system crontab -->
  <param name="systab" value="0"/>
  <!-- su, sudo or custom command, empty for none -->
  <param name="changeuser" value=""/>
  <param name="timeout" value="600"/>
</params>

<script language      = "java"
          java_class   = "sos.scheduler.cron.JobSchedulerCronAdapter"/>

<run_time let_run = "yes"
          repeat  = "120"
          begin   = "00:00"
          end     = "24:00"
          once    = "yes"/>

</job>
```

Zur Parametrisierung siehe Anhang (Seite 12)

3.1 Der Cron Adapter Job mit zentraler Konfiguration

Der Cron Adapter Job kann auch genutzt werden, um per zentraler Konfiguration cron Jobs von einem Supervisor Scheduler aus an Job Scheduler auf anderen hosts zu verteilen.

Dazu sind die folgenden Schritte nötig:

- Ein Job Scheduler muss als zentraler supervisor installiert werden
- Die Workload Job Scheduler müssen konfiguriert werden, sich bei diesem Supervisor anzumelden (Das Cron Paket wird von den Workload Schemulern nicht benötigt)
- Im Konfigurationsverzeichnis des Supervisors werden Verzeichnisse zur Konfiguration der Workload Scheduler angelegt. (Siehe Referenzdokumentation zum Job Scheduler, Kapitel "Zentrale Konfiguration durch einen Supervisor Scheduler")
- In der `scheduler.xml` Konfigurationsdatei wird zusätzlich zu `scheduler_cron.xml` auch `scheduler_cron_remote.xml` per `<base>` Element eingebunden.

Nun können in den Unterverzeichnissen der host-spezifischen Konfigurationsverzeichnisses `crontab` Dateien abgelegt werden, deren Jobs dann an die entsprechenden Job Scheduler verteilt werden. Es sollte dabei in jedem host-Verzeichnis ein spezielles Unterverzeichnis (z.B. `cron`) verwendet werden, da sonst Konfigurationsdateien, die sich direkt im host-Verzeichnis befinden, gelöscht werden.

Beispiel: crontabs für Job Scheduler auf host2 und host3

Ein Job Scheduler auf host1 soll cron Jobs für host2 und host3 konvertieren und verteilen. Wenn alle Scheduler wie oben beschrieben konfiguriert sind, werden die Workload Scheduler mit crontab Dateien in den folgenden Pfaden konfiguriert:

```
config/remote/host2#4444/cron/crontab - enthält cron Jobs für host2
config/remote/host3#4444/cron/crontab - enthält cron Jobs für host3
config/remote/_all/cron/crontab - enthält cron Jobs für alle Workload Scheduler
```

Um die cron Verteilung nur für bestimmte Workload Scheduler zu aktivieren, kann in `scheduler_cron_remote.xml` der Parameter `cron_adapter_dynamic_configuration_dir` angepasst werden. Hier werden die Verzeichnisse (durch ; getrennt) angegeben, in denen (und in deren Unterverzeichnissen) nach `crontab` Dateien gesucht wird.

4 User crontab oder System crontab?

Es gibt zwei Arten von `crontab` Dateien: `User crontab` und `System crontab`.

Eine `User crontab` Datei beinhaltet 5 Spalten für die Konfiguration des Intervals und eine Spalte für das auszuführende Kommando.

Eine `System crontab` Datei beinhaltet 5 Spalten für die Konfiguration des Intervals, eine Spalte für den Benutzernamen, unter dem das Kommando ausgeführt werden soll, und eine Spalte für das auszuführende Kommando.

Dem *Cron Converter* und dem Adapter Job muss daher mit dem Parameter `sysstab` mitgeteilt werden, ob es sich bei der einzulesenden `crontab` Datei um eine `User crontab` oder um eine `System crontab` handelt.

Bei einer `User crontab` werden die enthaltenen Befehle von dem Benutzer ausgeführt, unter dem der Job Scheduler gestartet wurde.

Bei einer `System crontab` kann der Benutzer gewechselt werden. Per Voreinstellung führt der Job Scheduler das Kommando unter dem Benutzer aus, der den Job Scheduler gestartet hat.

Sollen die Kommandos mit den in der `System crontab` angegebenen Benutzern ausgeführt werden, so muss mit dem Parameter `changeuser` ein Kommando für den Benutzerwechsel gewählt werden. Steht `changeuser` auf `su`, so führt der Job Scheduler das Kommando mit

```
su $SCHEDULER_CRONTAB_USER -c Kommando
```

aus. (Die Umgebungsvariable `$SCHEDULER_CRONTAB_USER` enthält den Benutzernamen aus der 6. Spalte der `System crontab`). Dies funktioniert nur, wenn der Job Scheduler unter dem `root` Benutzer ausgeführt wird. Wird er unter einem anderen Benutzer ausgeführt, dann erzeugt das `su` Kommando eine Kennwortabfrage, die eine Eingabe erwartet und nicht vom Job Scheduler im Batch-Betrieb beantwortet werden kann.

Soll der Job Scheduler unter einem anderen Benutzer als `root` laufen und trotzdem die Kommandos der `System crontab` ausführen, so kann `sudo` verwendet werden. Dazu muss der Parameter `changeuser` auf den Wert `sudo` gesetzt werden. Ausserdem muss `sudo` mit Hilfe der `/etc/sudoers` Datei so konfiguriert werden, dass der Benutzer, der den Job Scheduler ausführt, die geplanten Befehle unter den Konten der entsprechenden anderen Benutzer ausführen darf.

Beispiel: Scheduler User 'scheduler' soll ls als user 'test' ausführen dürfen

```
scheduler localhost = (test) /bin/lis NOPASSWD
```

Sind keine `sudo` Kenntnisse vorhanden, wird empfohlen, zunächst ein `sudo` Tutorial oder die *sudoers manpage* zu lesen.

Anhang A: Parameter

Das *Cron Converter* Kommandozeilen Tool und Der *Cron Adapter* Job werden mit den folgenden Parametern konfiguriert:

Converter Parameter	Job Parameter	Beschreibung
-crontab	crontab	Pfad und Dateiname der <code>crontab</code> Datei
-target	---	Zieldatei im XML-Format mit dem Konvertierungsergebnis
-systab	systab	Stellt ein, ob die aktuelle <code>crontab</code> Datei als <code>System crontab</code> oder als <code>User crontab</code> behandelt werden soll. Mögliche Werte: <ul style="list-style-type: none"> • <code>0</code>: User <code>crontab</code> • <code>1</code>: System <code>crontab</code> Die Voreinstellung ist <code>0</code> . Wenn als <code>crontab</code> Parameter <code>"/etc/crontab"</code> angegeben wird, so ist die Voreinstellung <code>1</code> .
-changeuser	changeuser	Wählt ein Kommando, um bei Verwendung einer <code>System crontab</code> den Benutzer zu wechseln. Mögliche Werte: <ul style="list-style-type: none"> • <code>su</code>: Ausführung des Kommandos mit <code>su</code> • <code>sudo</code>: Ausführung des Kommandos mit <code>sudo</code> • Es kann auch ein eigenes Kommando (unter Verwendung von <code>\$(SCHEDULER_CRONTAB_USER)</code>) angegeben werden. Wird kein Kommando gewählt, so wird der Benutzer nicht gewechselt.
-timeout	timeout	Gibt einen Timeout (in Sekunden) für die Ausführung der Jobs an. (Voreinstellung: 600)
-v	---	loglevel [0=info] [1=debug1]...[9=debug9]

Für eine Beispielkonfiguration des Jobs siehe Kapitel 'Der Cron Adapter Job' .

A.1 Steuern der Job-Konvertierung

Mit Hilfe von Kommentaren kann in der `crontab` Datei auf das Ergebnis der Konvertierung Einfluss genommen werden. Job-Name, Job-Titel und Job-Timeout lassen sich pro Job über Kommentare setzen. Dazu muss sich vor dem jeweiligen Job ein Kommentar in einer der Formen:

- `# job_name` = Name des Jobs, andernfalls wird der Name aus dem Kommando gebildet
- `# job_title` = Titel des Jobs
- `# job_timeout` = Timeout (in Sekunden) des Jobs, andernfalls gilt die Voreinstellung

finden.

Beispiel: Job soll `my_cron_job` heissen

```
# job_name = my_cron_job
# other comment(s)
0 * * * * ls -la
```

Anhang B: Einschränkungen

Die 5 Spalten einer `crontab` für die Laufzeitkonfiguration bieten eine große Anzahl von Kombinationsmöglichkeiten. Ein Großteil davon kann durch den *Cron Converter* konvertiert werden, jedoch gibt es für einige Kombinationen Einschränkungen, die hier benannt werden sollen:

Cron-Runtime	Bemerkung
<code>* /2 * /3 * * *</code>	Ein Wiederholungsintervall ist für Minuten und für Stunden angegeben. In diesem Fall soll der Job also alle 3 Stunden in einem Intervall von 2 Minuten wiederholt werden. Da diese Kombination unsinnig erscheint, wird sie nicht unterstützt.
<code>0 10 * /3 * *</code>	Der Job soll alle drei Tage um 10 Uhr wiederholt werden. Wiederholungsintervalle über Tage werden zwar nicht unterstützt, jedoch generiert der Converter eine Laufzeit mit festgesetzten Monatstagen, die den entsprechenden Abstand haben. In diesem Fall also 1,4,7,10 ...
<code>0 10 1 * /3 *</code>	Der Job soll alle drei Monate am 1. um 10 Uhr wiederholt werden. Hier gilt dasselbe wie bei Wiederholungsintervallen für Tage. Es werden festgesetzte Monate mit entsprechendem Abstand generiert.
<code>0 10 * * * /3</code>	Der Job soll alle drei Wochentage 10 Uhr laufen. Wiederholungsintervalle für Wochentage werden nicht unterstützt. Es können Wiederholungsintervalle in der 3. Spalte (Monatstage) stattdessen verwendet werden.
<code>0 10 13 * 5</code>	Der Job soll jeden Freitag den 13. um 10 Uhr laufen Die Verknüpfung eines bestimmten Wochentags mit einem bestimmten Monatstag wird nicht unterstützt. Die Wochentagsspalte wird nur ausgewertet, wenn die Monatstagsspalte auf * gesetzt ist.