

JOB SCHEDULER

Kompilieren des Job Schedulers

Technische Dokumentation

9. April 2008

Inhaltsverzeichnis

1	Voraussetzungen.....	3
2	Auslieferung des Job Scheduler Source Codes	3
3	Vorbereitung	4
3.1	Auspacken der Quellen	4
3.2	Anlegen von symbolischen Links.....	4
3.2.1	Symbolic Link aufs Java-SDK	4
3.2.2	Symbolic Link auf eine Perl-Installation.....	4
3.3	Java Archive	4
4	Das Kompilieren	5

1 Voraussetzungen

Um den Job Scheduler zu kompilieren, müssen folgende Voraussetzungen erfüllt sein:

- GNU Compiler `gcc` Version 4.1.2
- `gdb` (GNU Make Debugger)
- Java SDK Version 1.4, 1.5
- `Bzip2`
- Perl 5.8.0 – 5.8.8 (falls die Perl API Unterstützung vom Scheduler gewünscht ist)
- `ant` und `javadoc` (nur für Dokumentation der Java-Schnittstellen)
- `unixODBC-devel` (Dateien `sql.h` und `sqlext.h`)
- `zlib-devel`
- `gcc41-32bit` und `libstdc++41-devel-32bit` (erforderlich, wenn das Kompilieren auf x86_64 Linux für 32-bit Systeme erfolgt)
- `binutils` (passende Version zu `gcc`)
- `lex` (Linux; `flex`), `ar`

Für den Betrieb des Job Schedulers sind folgende Komponenten erforderlich bzw. wünschenswert

- DBMS: Oracle 8.1.7, 9.2, 10g; DB2 8.x, 9.x; SQL Server 2000, 2005; MySQL 4.1.x, 5.x; PostgreSQL 8.x; Firebird 1.5, 2.x
- SMTP Server
- Web Server (IIS, Apache), PHP 5.x, PHP Extensions für das jeweilige DBMS

2 Auslieferung des Job Scheduler Source Codes

Ausgeliefert werden die Quellen komprimiert im BZIP2-Format. Zum Herunterladen steht Ihnen die Datei `scheduler_sorces.tar.bz2` zur Verfügung unter:

http://sourceforge.net/project/showfiles.php?group_id=145323

oder

http://jobscheduler.sourceforge.net/osource_scheduler_download_de.php

oder

http://www.sos-berlin.com/modules/cjaycontent/index.php?id=osource_scheduler_download_de.php

3 Vorbereitung

3.1 Auspacken der Quellen

Nach dem die Quellen heruntergeladen wurden, müssen sie entpackt werden. Dies geschieht am besten mit dem Befehl:

```
bzip2 <scheduler_sources.tar.bz2 | tar xf -
```

Die heruntergeladenen Dateien werden innerhalb des aktuellen Arbeitsverzeichnisses entpackt. Dabei wird ein neues Verzeichnis mit dem Namen `scheduler` angelegt.

3.2 Anlegen von symbolischen Links

Um einige Voraussetzungen für das Kompilieren zu erfüllen, sind ein paar symbolische Links notwendig. Dafür müssen Sie ins Verzeichnis `LINKS` wechseln

```
cd scheduler/prod/LINKS
```

3.2.1 Symbolic Link aufs Java-SDK

Hier muss ein Link aufs Java-SDK gesetzt werden:

Beispiel: `ln -s /opt/java java`

3.2.2 Symbolic Link auf eine Perl-Installation

Außerdem wird ein Verweis auf Ihre Perl-Installation benötigt, falls die Perl API Unterstützung vom Scheduler gewünscht ist:

Beispiel: `ln -s /opt/perl/lib/5.8.0/i686-linux-thread-multi perl`

3.3 Java Archive

Kopieren Sie die Java Archive **mail.jar**, **xercesImpl.jar** und **activation.jar** ins Verzeichnis `prod/LINKS` (oder setzen Sie einen Link).

4 Das Kompilieren

Um den Job Scheduler zu kompilieren, wechseln Sie ins Verzeichnis `prod`:

```
cd prod
```

Dann starten Sie das Skript `make-scheduler`:

```
make/make-scheduler
```

Das Scheduler Perl-Modul muss auch kompiliert werden, falls die Perl API Unterstützung vom Scheduler gewünscht ist:

```
make/make-libsosperlscript
```

Um JavaScript zu kompilieren:

```
cd spidermonkey
```

```
make
```

Nachdem das Kompilieren erfolgreich abgeschlossen ist, stehen Ihnen die folgenden Dateien zur Verfügung:

File	Beschreibung
prod/bin/scheduler	Der Job Scheduler
prod/bin/sos.spooler.jar	Basisklassen für die Implementierung von Java Jobs
prod/bin/sos.mail.jar	eMail-Versand mit JavaMail
spidermonkey/bin/libspidermonkey.so	Basisklassen für die Implementierung von JavaScript Jobs