



JOB SCHEDULER

How To Compile Job Scheduler

Technical Information

April 9, 2008

Table of Contents

1	Prerequisites	3
2	Distribution of Job Scheduler Source Code	3
3	Preparations	4
3.1	Extract sources	4
3.2	Create Symbolic Links	4
3.2.1	Symbolic Link for Java SDK	4
3.2.2	Symbolic Link for Perl Installation	4
3.3	Java Archives	4
4	Compiling	5

1 Prerequisites

In order to compile Job Scheduler the following prerequisites have to be fulfilled:

- GNU Compiler `gcc` starting with version 4.1.2
- `gdb` (GNU Make Debugger)
- Java SDK starting with version 1.4, 1.5
- `Bzip2`
- Perl 5.8.0 – 5.8.8 (if you plan to use the Perl API support)
- `ant` and `javadoc` (documentation of the Java API)
- `unixODBC-devel` (files `sql.h` and `sqlext.h`)
- `zlib-devel`
- `gcc41-32bit` and `libstdc++41-devel-32bit` (required if you compile on a `x86_64` Linux system for 32-bit system)
- `binutils` (version corresponding to `gcc`)
- `lex` (Linux; `flex`), `ar`

For operation of the Job Scheduler the following components are required/desirable:

- 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 for the respective DBMS

2 Distribution of Job Scheduler Source Code

The distribution contains sources that are compressed in a BZIP2 format. The download is available in the download section of the web site and in the archive `scheduler_sorces.tar.bz2` at:

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

or

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

or

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

3 Preparations

3.1 Extract sources

After download of the source archive the source files have to be extracted. You could use the command:

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

The files will be extracted to the current working directory. A new directory `scheduler` will be created.

3.2 Create Symbolic Links

Some symlinks are required for a successful compilation. Switch to the `LINKS` directory:

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

3.2.1 Symbolic Link for Java SDK

A symlink for your JavaSDK is required:

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

3.2.2 Symbolic Link for Perl Installation

A symlink for your Perl installation is required, if you plan to use the Perl API support:

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

3.3 Java Archives

Copy the Java archives **mail.jar**, **xercesImpl.jar** and **activation.jar** to the directory `prod/LINKS` (alternatively create a symlink).

4 Compiling

Switch to the directory `prod` in order to compile Job Scheduler:

```
cd prod
```

Then run the script `make-scheduler`:

```
make/make-scheduler
```

The Perl-Modul that comes with the Scheduler should also be compiled, if you plan to use the Perl API support:

```
make/make-libsosperlscript
```

In order to compile JavaScript use:

```
cd spidermonkey  
make
```

After successful compilation the following files will be available:

File	Description
<code>prod/bin/scheduler</code>	The Job Scheduler
<code>prod/bin/sos.spooler.jar</code>	Base Classes for Java job implementations
<code>prod/bin/sos.mail.jar</code>	Mail connector class for JavaMail
<code>spidermonkey/bin/libspidermonkey.so</code>	Base Classes for JavaScript job implementation