



**JOB**SCHEDULER

# JobScheduler - Installation und Konfiguration

Installation and Configuration  
Juli 2013

## Impressum

Software- und Organisations-Service GmbH

Giesebrechtstr. 15  
D-10629 Berlin  
Germany

Telefon +49 (0)30 86 47 90-0  
Telefax +49 (0)30 8 61 33 35  
Mail [info@sos-berlin.com](mailto:info@sos-berlin.com)  
Web <http://www.sos-berlin.com>

Letzte Aktualisierung: 07/10/2013 05:02 PM

Diese Dokumentation basiert auf der JobScheduler Version 1.5.3191.

**Copyright © 2005-2013 SOS GmbH Berlin.**

All rights reserved. All trademarks or registered trademarks are the property of their respective holders. Alle Informationen in diesem Dokument können ohne vorherige Ankündigung verändert werden.

This product includes software developed by the Apache Software Foundation (<http://apache.org/>)

We would appreciate any feedback you have, or suggestions for changes and improvements; please forward your comments to [info@sos-berlin.com](mailto:info@sos-berlin.com).

# Inhaltsverzeichnis

- 1 Installation ..... 5**
  - 1.1 Voraussetzungen ..... 5
    - 1.1.1 Voraussetzungen für den 32-Bit JobScheduler ..... 5
    - 1.1.2 Voraussetzungen für den 64-Bit JobScheduler ..... 6
  - 1.2 Installation per Setup-Programm ..... 7
  - 1.3 Lizenzen ..... 8
  - 1.4 Installationspfade ..... 9
  - 1.5 Pakete des Setups ..... 11
  - 1.6 Formulare des Setups ..... 12
    - 1.6.1 Formulare des Basis-Pakets JobScheduler ..... 13
    - 1.6.2 Formular des Pakets Update Service ..... 16
    - 1.6.3 Formulare des Pakets Database Support ..... 17
    - 1.6.4 Formular des Pakets Cron Job ..... 21
- 2 Batch-Installation ..... 22**
- 3 Einrichten der Datenbank ..... 27**
  - 3.1 MySQL ..... 27
  - 3.2 MS SQL Server and Sybase ..... 27
  - 3.3 PostgreSQL ..... 27
  - 3.4 Manuelles Einrichten der Tabellen ..... 28
- 4 Verzeichnisstruktur nach der Installation ..... 29**
- 5 Starten und Stoppen des JobScheduler ..... 33**
  - 5.1 JobScheduler Demon für Unix ..... 33
  - 5.2 JobScheduler Dienst für Windows ..... 33
- 6 JOC, JOE und JID öffnen ..... 35**
  - 6.1 JOC (JobScheduler Operations Center) öffnen ..... 35
  - 6.2 JOE (JobScheduler Object Editor) öffnen ..... 35
  - 6.3 JID (JobScheduler Information Dashboard) öffnen ..... 36
- 7 Konfiguration ..... 37**
  - 7.1 Die Datei sos.ini ..... 37
  - 7.2 Die Datei factory.ini ..... 37
  - 7.3 Die Datei scheduler.xml ..... 37
  - 7.4 Die Datei custom.js ..... 37
  - 7.5 Die Datei hibernate.cfg.xml ..... 38
  - 7.6 Die Datei jobscheduler\_environment\_variables.(sh|cmd) ..... 38
  - 7.7 Die Datei jobeditor\_environment\_variables.(sh|cmd) ..... 38
  - 7.8 Die Datei dashboard\_environment\_variables.(sh|cmd) ..... 38
- 8 JobScheduler aktualisieren ..... 40**
- 9 Mehrfache Installationen ..... 41**
- 10 Installation eines Clusters ..... 42**
- 11 Deinstallation ..... 43**
  - 11.1 Deinstallation per Uninstaller ..... 43
  - 11.2 Manuelle Deinstallation unter Windows ..... 43
  - 11.3 Manuelle Deinstallation unter Unix ..... 44

---

<b>12 Verfahren des automatischen Updates .....</b>	<b>45</b>
<b>13 Betrieb des 32bit JobScheduler auf 64bit Systemen .....</b>	<b>46</b>
13.1 32bit JOE and JID auf 64bit Systemen .....	46
<b>14 Troubleshooting .....</b>	<b>47</b>
<b>15 Glossar .....</b>	<b>48</b>

# 1 Installation

Bei einer Neuinstallation des **JobScheduler** sollte in folgender Reihenfolge vorgegangen werden:

## Einrichten der Datenbank (Seite 27) (optional, empfohlen)

Der **JobScheduler** kann auch ohne eine Datenbank genutzt werden jedoch wird dringend empfohlen den **JobScheduler** mit Datenbank zu betreiben. Der **JobScheduler** braucht die Datenbank zum Archivieren der Protokolle und um sich Zustände der *Jobs* und *Aufträge* zu merken.

Das **JID (JobScheduler Information Dashboard)** benötigt zwingend eine Datenbank.

Unterstützte Datenbanksysteme sind MariaDB (1.1.3 +), MySQL 5.x, Oracle (8.1.7, 9.2, 10g, 11g), Microsoft SQL Server (2000, 2005), PostgreSQL (8.x, 9.x), Firebird 1.5, DB2 8.x und Sybase ASE 15.0.

Speziell für den Betrieb mit MySQL, MS SQL Server und Sybase müssen Sie aus lizenzrechtlichen Gründen selbst einen JDBC-Treiber bereitstellen, der mit Ihrer verwendeten MySQL, MS SQL Server bzw. Sybase Version korrespondiert. Alternativ kann für MS SQL Server und Sybase der mitgelieferte jTDS JDBC-Treiber verwendet werden. Für Oracle, PostgreSQL, Firebird und DB2 wird dieser vom Setup mitgeliefert.

## JobScheduler Installation (Seite 7)

Die Installation erfolgt mittels Setup-Programm. Für Windows und Linux können Sie das unter <http://www.sos-berlin.com> heruntergeladen. Für Solaris, HP-UX (Itanium) und AIX erhalten Sie den **JobScheduler** auf Anfrage.

Supported operation systems are Windows 2003/XP(SP2)/Vista/2008/7, Linux starting with kernel 2.4, Solaris sparc 8/9/10 and AIX 6/7.

## 1.1 Voraussetzungen

Für Linux und Windows ist der **JobScheduler** in 64-Bit und 32-Bit verfügbar, für andere Betriebssysteme zur Zeit noch ausschließlich in 32-Bit.

### 1.1.1 Voraussetzungen für den 32-Bit JobScheduler

- Oracle™ Java Runtime Environment (JRE) 32-Bit ab Version 1.6.x Für AIX darf auch das IBM™ Java 32-Bit ab Version 1.6.x verwendet werden.
- Für Unix:

Eine Shell in /bin/sh (oder ein Symlink)

Einige abhängige 32-Bit Bibliotheken sind für den **JobScheduler** erforderlich. Diese sind auf Linux:

- `linux-gate.so.1`
- `libz.so.1`
- `libpthread.so.0`
- `libdl.so.2`
- `libm.so.6`
- `ld-linux.so.2`

Zwei Komponenten (**JOE (JobScheduler Object Editor)** and **JID (JobScheduler Information Dashboard)**) des **JobScheduler** sind SWT Anwendungen welche ein X-Windows System und GTK2 benötigen. Die Installation enthält ein 32-Bit swt.jar, somit brauchen Sie folgende 32-Bit Bibliotheken:

- `libgtk-x11-2.0.so.0`
- `libXtst.so.6`
- Für Windows:
 

Bei Verwendung von Java 1.6 muss die Bibliothek `msvcr71.dll` in `C:\Windows\system32` und ebenfalls auf 64-Bit Windows in `C:\Windows\SysWOW64` vorliegen.

Bei Verwendung von Java 1.7 muss die Bibliothek `msvcr100.dll` in `C:\Windows\system32` und ebenfalls auf 64-Bit Windows in `C:\Windows\SysWOW64` vorliegen.

Sie finden die `msvcr71.dll` oder `msvcr100.dll` in der Java Installation `[jre install path]\bin\plugin2`.
- Bei Verwendung der "Remote Configuration" sollten die Workload **JobScheduler** und ihr Supervisor **JobScheduler** die gleiche Version haben. Weitere Informationen über "Remote Configuration" finden Sie [here](#).

### 1.1.2 Voraussetzungen für den 64-Bit JobScheduler

- Oracle™ Java Runtime Environment (JRE) 64-Bit ab Version 1.6.x Für AIX darf auch das IBM™ Java 64-Bit ab Version 1.6.x verwendet werden.
- Für Unix:

Eine Shell in `/bin/sh` (oder ein Symlink)

Einige abhängige 64-Bit Bibliotheken sind für den **JobScheduler** erforderlich. Diese sind auf Linux:

- `linux-vdso.so.1`
- `libz.so.1`
- `libpthread.so.0`
- `libdl.so.2`
- `libm.so.6`
- `libc.so.6`
- `ld-linux-x86-64.so.2`

Zwei Komponenten (**JOE (JobScheduler Object Editor)** and **JID (JobScheduler Information Dashboard)**) des **JobScheduler** sind SWT Anwendungen welche ein X-Windows System und GTK2 benötigen. Die Installation enthält ein 64-Bit swt.jar, somit brauchen Sie folgende 64-Bit Bibliotheken:

- `libgtk-x11-2.0.so.0`
- `libXtst.so.6`
- Bei Verwendung der "Remote Configuration" sollten die Workload **JobScheduler** und ihr Supervisor **JobScheduler** die gleiche Version haben. Weitere Informationen über "Remote Configuration" finden Sie [here](#).

## 1.2 Installation per Setup-Programm

Die folgenden Archiv-Dateien inklusive Installer sind verfügbar

- `jobscheduler_linux-x64.[release].tar.gz` für Linux 64-Bit
- `jobscheduler_linux-x86.[release].tar.gz` für Linux 32-Bit
- `jobscheduler_windows-x64.[release].zip` für Windows 64-Bit
- `jobscheduler_windows-x86.[release].zip` für Windows 32-Bit
- `scheduler_solarisx86.[release].tar.gz` für Solaris
- `scheduler_solaris-sparc.[release].tar.gz` für Solaris
- `scheduler_hpux-ia64-32.[release].tar.gz` für HP-UX Itanium
- `scheduler_aix32.[release].tar.gz` für AIX

Entpacken Sie das Archive in ein beliebiges Verzeichnis und wechseln Sie in das ausgepackte Verzeichnis `./jobscheduler.[release]`

Das Setup-Programm kann als Dialog und im Batch (siehe [Batch-Installation](#) (Seite 22)) gestartet werden. Wird das Setup unter Unix als Dialog-Programm aufgerufen, dann muss ein X-Server installiert sein. Sollte kein X-Server installiert sein, so nutzen Sie bitte die [Batch-Installation](#).

```
/tmp/jobscheduler.[release]> setup.sh
```

Beispiel: Installationsprogramm auf Unix starten

```
C:\Windows\Temp\jobscheduler.[release]>setup.cmd
```

Beispiel: Installationsprogramm auf Windows starten

Für das Setup sind unter Windows Administrator-Rechte erforderlich. Unter Windows öffnet das Setup gegebenenfalls einen entsprechenden Dialog. In Unix-Umgebungen wird ein sudo-Prompt geöffnet. Starten Sie das Setup unter Unix nicht als root, sondern benutzen Sie sudo!

Unter Unix kann auf die root-Rechte verzichtet werden. Wenn Sie den Job Scheduler ohne root-Rechte installieren möchten, dann rufen Sie

```
/tmp/jobscheduler.[release]> setup.sh -u
```

Beispiel: Installationsprogramm auf Unix ohne root-Rechte starten

Der Setup-Dialog startet, in dem Sie als erstes die Sprache des Setup wählen, nach einer Begrüßung die Lizenzbestimmungen bestätigen und zwei Installationspfade auswählen. Unterhalb des ersten Pfads werden die Programmdateien und Bibliotheken gespeichert. Unterhalb des zweiten Pfads werden die Konfigurations- und Protokolldateien gespeichert.

Im weiteren wird innerhalb dieser Dokumentation der erste Installationspfad `$_SCHEDULER_HOME` und der zweite `$_SCHEDULER_DATA` genannt. Anschließend sehen Sie den Dialog der [Paketauswahl](#) (Seite 11).

Abhängig von der Wahl der Pakete bieten weitere Formulare Konfigurationsmöglichkeiten. Näheres hierzu ist im Kapitel [Formulare des Setups](#) (Seite 12) beschrieben. Dann werden die Dateien der gewählten Pakete kopiert und diverse Skripte ausgeführt, um die Installation ihrer Konfiguration anzupassen. Der Verlauf der vom Setup

aufgerufenen Skripte wird geloggt. Sie finden die Log-Datei unter `$_SCHEDULER_DATA/logs` mit dem Namen `Instal1_V[release]_[datum][uhrzeit]_[laufnr].log`.

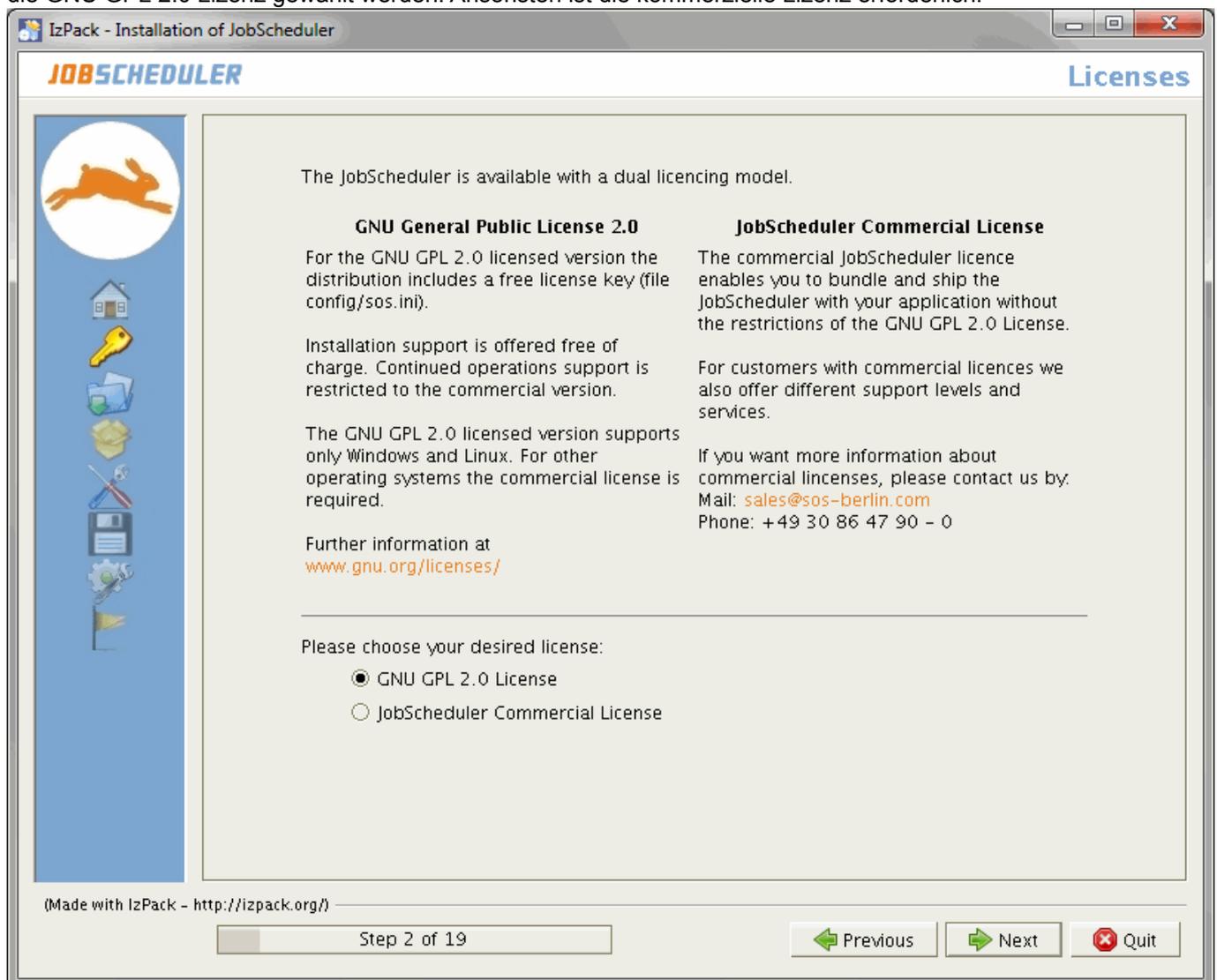
Am Ende des Setups können Sie **JOC (JobScheduler Operations Center)** bereits aufrufen, in dem Sie in Ihrem Browser (unterstützt werden Internet Explorer und Firefox) als URL

`http://localhost:[port]`

eingeben, wobei [port] die innerhalb des Setup von Ihnen gemachte Angabe zum Port des **JobScheduler** ist.

### 1.3 Lizenzen

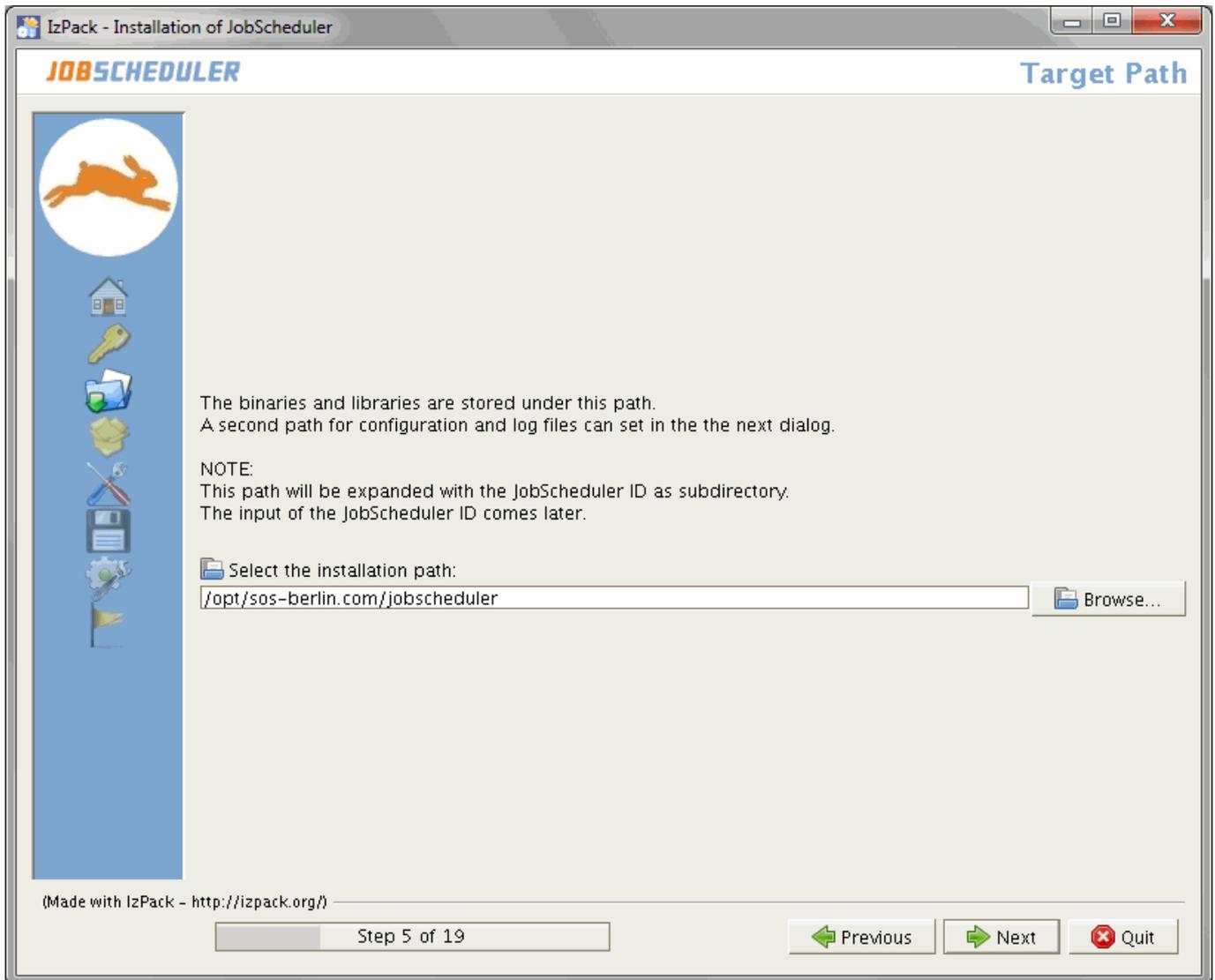
Der Job Scheduler ist mit zwei verschiedenen Lizenzen verfügbar (dual licensing). Für Windows und Linux kann die GNU GPL 2.0 Lizenz gewählt werden. Ansonsten ist die kommerzielle Lizenz erforderlich.



Wird die kommerzielle Lizenz gewählt, so wird ein Eingabefeld für den Lizenzschlüssel erscheinen. Der Lizenzschlüssel wird in die Datei `$_SCHEDULER_DATA/config/sos.ini` geschrieben. Sollte der Lizenzschlüssel ungültig sein, kann die Installation trotzdem fortgeführt werden und später in die `$_SCHEDULER_DATA/config/sos.ini` eingefügt werden.

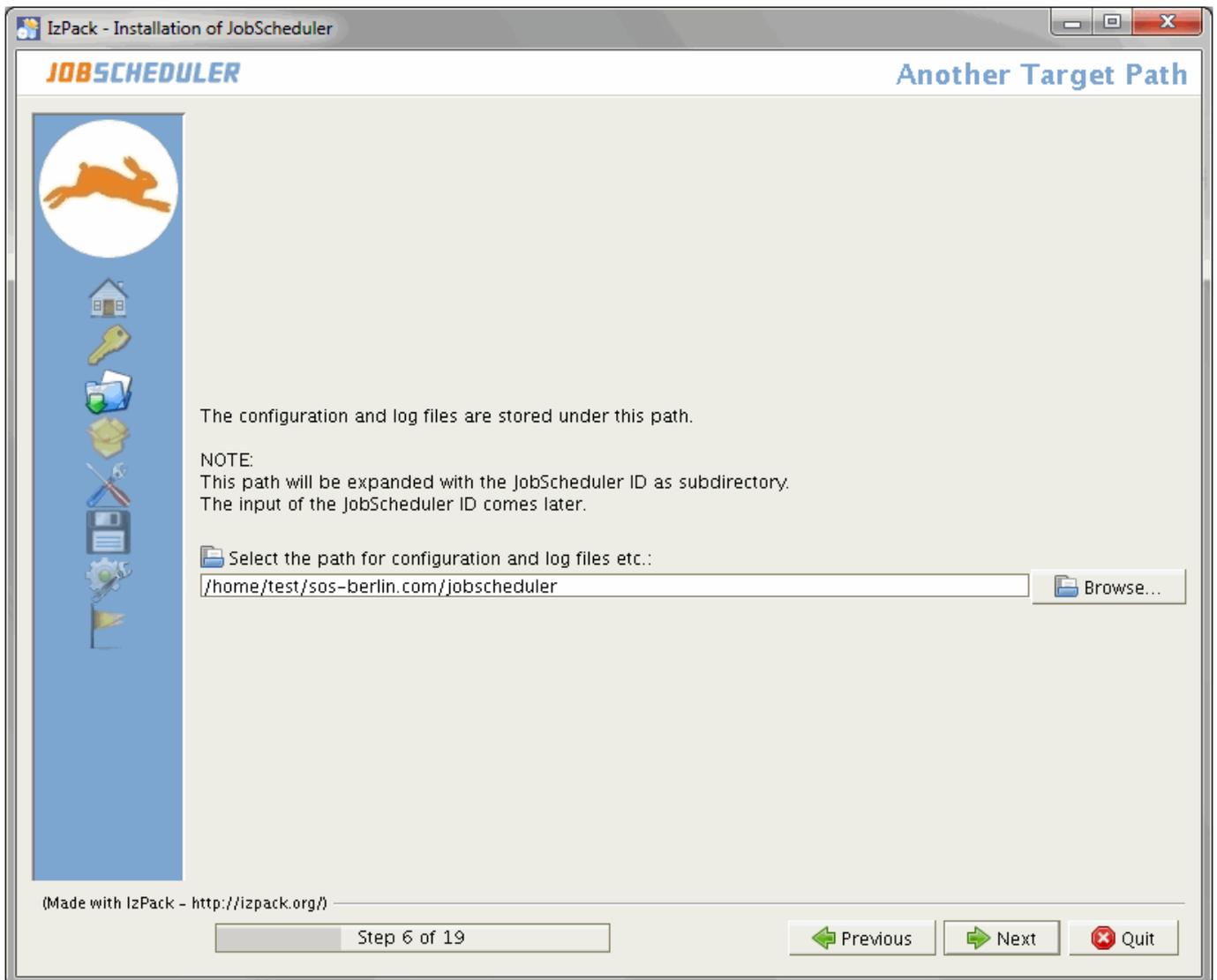
## 1.4 Installationspfade

Das Setup kennt zwei Pfade. Beide Pfade werden durch die später anzugebene **JobScheduler** ID als Unterverzeichnis ergänzt.



In diesem ersten Installationspfad (`$SCHEDULER_HOME`) werden die Programmdateien und Bibliotheken gespeichert. Der voreingestellte Pfad ist

- `/opt/sos-berlin.com/jobscheduler` für Unix. Wenn Sie das Setup ohne root-Rechte gestartet haben, dann müssen Sie hier ein anderes Verzeichnis wählen (z.B. `/home/[user]/sos-berlin.com/jobscheduler`).
- `C:\Program Files\sos-berlin.com\jobscheduler` für Windows



In diesem zweiten Installationspfad (`$SCHEDULER_DATA`) werden die Konfigurations- und Protokolldateien gespeichert. Der voreingestellte Pfad ist

- `/home/[user]/sos-berlin.com/jobscheduler` für Unix
- `C:\ProgramData\sos-berlin.com\jobscheduler` für Windows Vista/2008/7 etc.
- `C:\Documents and Settings\All Users\Application Data\sos-berlin.com\jobscheduler` für ältere Windows

## 1.5 Pakete des Setups

Während des Setups können folgende Pakete ausgewählt werden

### JobScheduler

Das ist die Basis-Installation. Diese kann nicht abgewählt werden. In diesem Paket ist **JOC (JobScheduler Operations Center)** enthalten. Dies ist eine Web-Oberfläche zur Überwachung und Steuerung der **JobScheduler** Objekte, wie *Jobs*, *Job-Ketten* und *Aufträge*. Desweiteren enthält das Paket **JOE (JobScheduler Object Editor)** mit dem man die **JobScheduler** Objekte konfiguriert und **JID (JobScheduler Information Dashboard)**, welches eine Übersicht über die geplanten und tatsächlich gelaufenen *Jobs* zur Verfügung stellt.

### Update Service

Dem **JobScheduler** wird ein *Job* hinzugefügt, der wöchentlich prüft, ob es ein neues Release gibt.

### Database Support

Die *Job*-Protokolle können in einer Datenbank gespeichert werden. Unterstützte Datenbanksysteme sind MySQL 5.x, Oracle (8.1.7, 9.2, 10g, 11g), Microsoft SQL Server (2000, 2005), PostgreSQL (8.x, 9.x), Firebird 1.5, DB2 8.x und Sybase ASE 15.0.

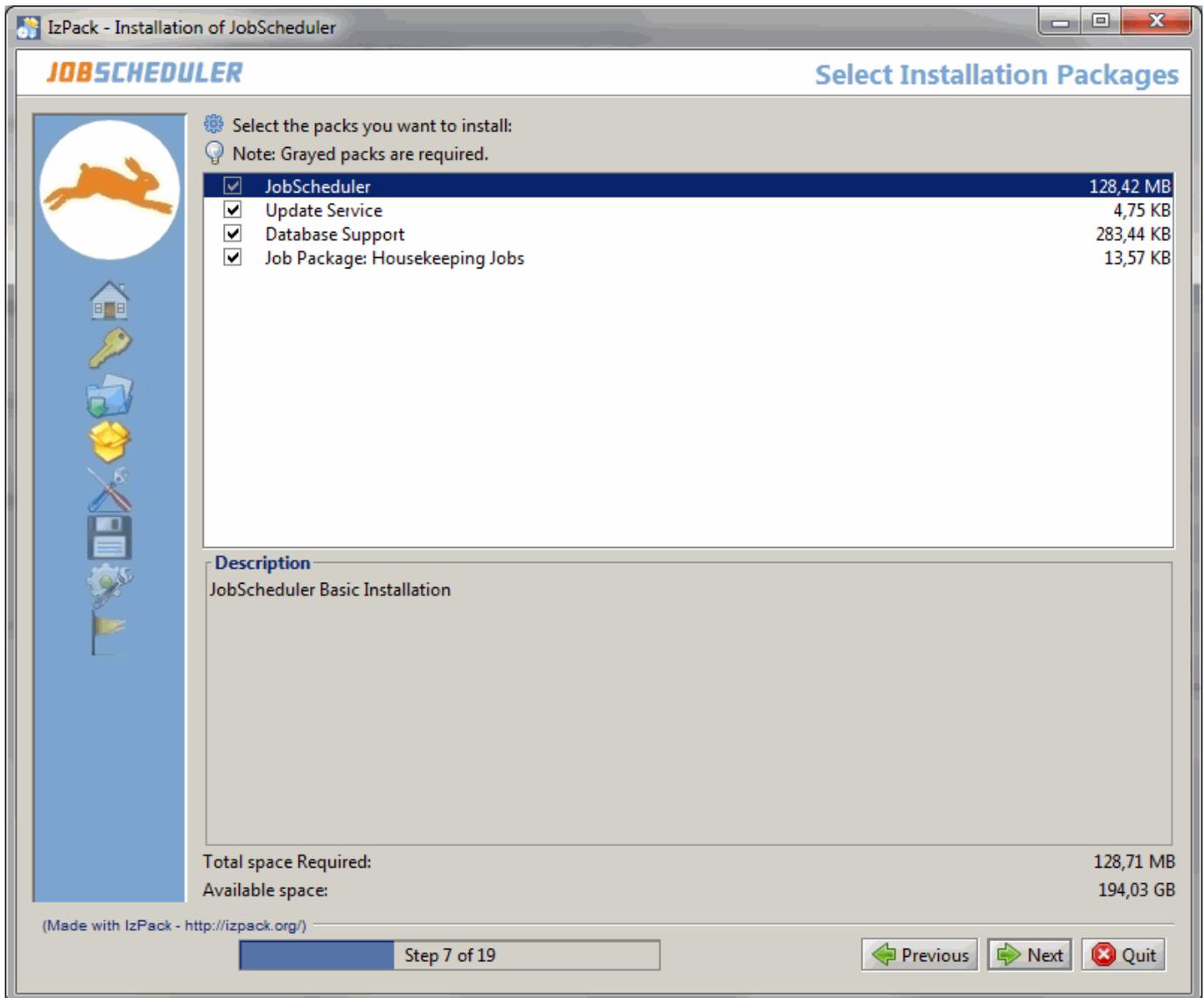
### Housekeeping Jobs

Housekeeping *Jobs* werden automatisch vom **JobScheduler** ausgeführt, bspw. für den Versand zwischengespeicherter Protokoll-eMails, das Entfernen temporärer Dateien oder den Neustart des **JobScheduler**. Außerdem kann der **JobScheduler** als Event Handler konfiguriert werden.

### Cron Job

Der Cron Adapter Job kann genutzt werden, um den **JobScheduler** mit einer crontab-Datei zu konfigurieren. Zu diesem Zweck liest der *Job* die crontab-Datei und konfiguriert den **JobScheduler** dynamisch. Dieses Paket ist nur unter Unix verfügbar.

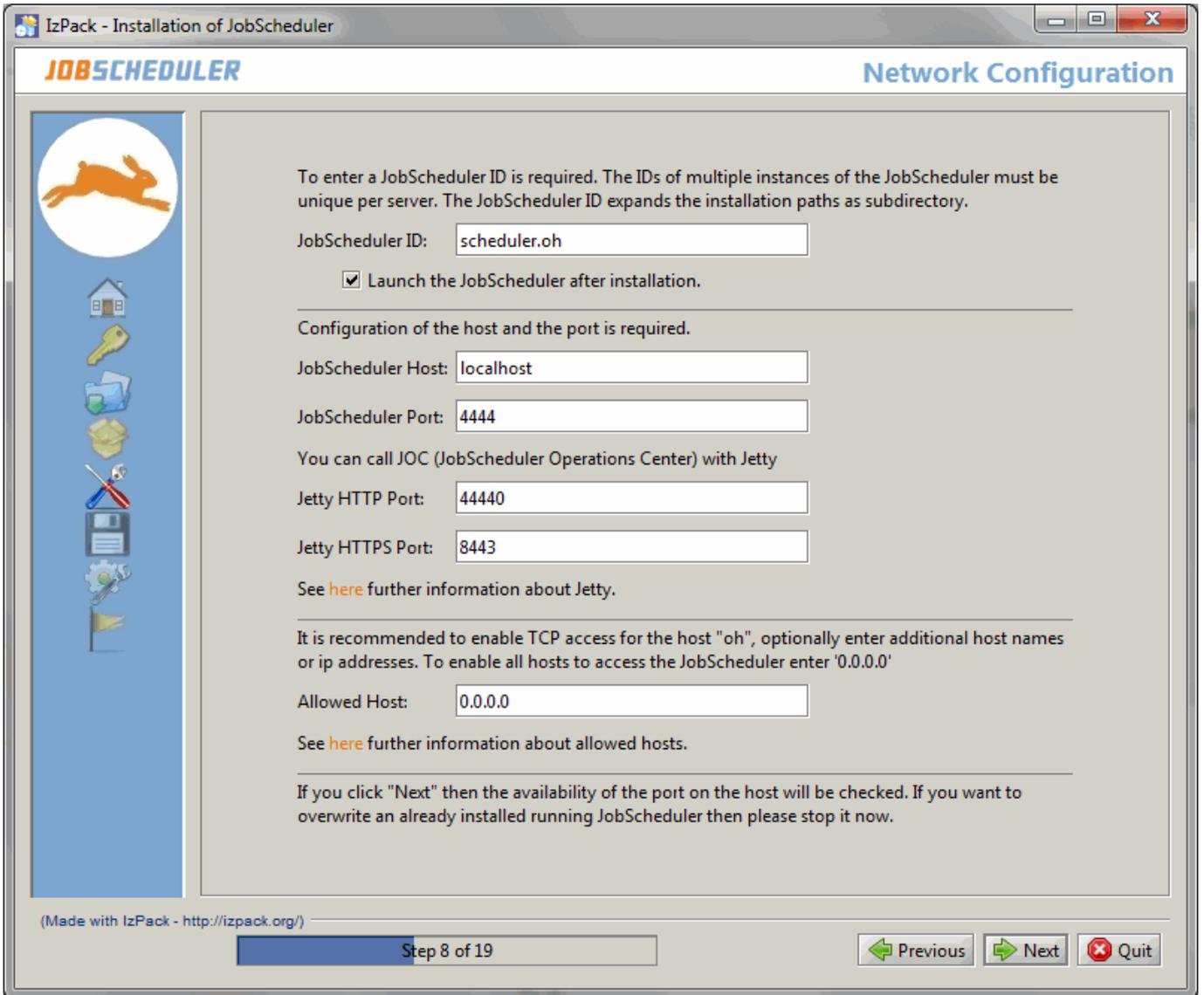
Hierzu wird während des Setups folgender Dialog angezeigt.



## 1.6 Formulare des Setups

Die Anzahl der während des Setups angezeigten Formulare ist abhängig von der Wahl der Pakete.

## 1.6.1 Formulare des Basis-Pakets JobScheduler



**JOB SCHEDULER** Network Configuration

To enter a JobScheduler ID is required. The IDs of multiple instances of the JobScheduler must be unique per server. The JobScheduler ID expands the installation paths as subdirectory.

JobScheduler ID:

Launch the JobScheduler after installation.

Configuration of the host and the port is required.

JobScheduler Host:

JobScheduler Port:

You can call JOC (JobScheduler Operations Center) with Jetty

Jetty HTTP Port:

Jetty HTTPS Port:

See [here](#) further information about Jetty.

It is recommended to enable TCP access for the host "oh", optionally enter additional host names or ip addresses. To enable all hosts to access the JobScheduler enter '0.0.0.0'

Allowed Host:

See [here](#) further information about allowed hosts.

If you click "Next" then the availability of the port on the host will be checked. If you want to overwrite an already installed running JobScheduler then please stop it now.

(Made with IzPack - <http://izpack.org/>)

Step 8 of 19

Previous Next Quit

Im Eingabefeld *JobScheduler ID* geben Sie bitte eine ID des **JobScheduler** an. Vermeiden Sie hierbei die folgenden Sonderzeichen / \ : ; \* ? ! \$ % & " < > ( ) | ^

Unter Windows bestimmt die ID zudem den Namen des nach dem Setup installierten Dienstes in der Form `jos_scheduler_[scheduler_id]`.

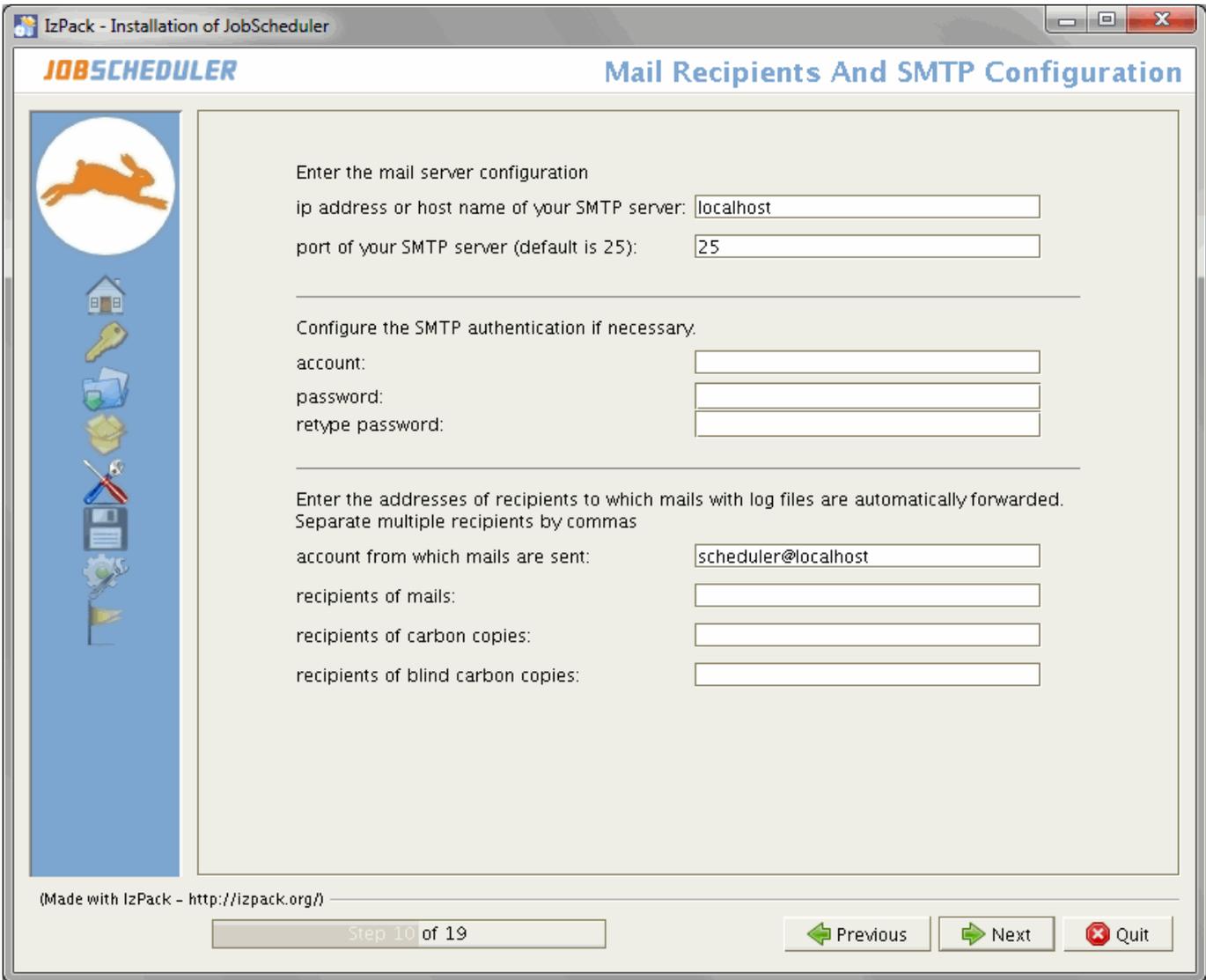
Die *JobScheduler ID* muss eindeutig über alle installierten **JobScheduler** gewählt werden (außer ein Cluster soll gebildet werden). Ferner ergänzt die *JobScheduler ID* die Installationspfade als Unterverzeichnis.

Die Angabe des *JobScheduler Ports* wird für die TCP-Kommunikation mit dem **JobScheduler** z.B. mit **JOC** benötigt.

**JOC** kann auch mit Jetty aufgerufen werden. Geben Sie für Jetty einen Port für HTTP und HTTPS an. Weitere Informationen zu Jetty finden Sie [hier](#).

Die *zugelassenen Rechner* genügen einem Sicherheits-Feature des **JobScheduler**, wonach dieser die Kommunikation auf bestimmte Rechner einschränken kann. Näheres hierzu entnehmen Sie bitte der [Dokumentation](#) zum **JobScheduler**.

Die Angaben zum *Port* und den *zugelassenen Rechner* sind konfiguriert in der Datei `$SCHEDULER_DATA/config/scheduler.xml`. Die Angaben zur **JobScheduler** ID ist konfiguriert in der Datei `$SCHEDULER_HOME/bin/jobscheduler_environment_variables.(cmd|sh)`. Die Angaben zu Jetty sind konfiguriert in der Datei `$SCHEDULER_DATA/config/jetty.xml` Die Dateien können nachträglich [manuell geändert](#) (Seite 37) werden.



**JOBSCHEDULER** Mail Recipients And SMTP Configuration

Enter the mail server configuration

ip address or host name of your SMTP server:

port of your SMTP server (default is 25):

---

Configure the SMTP authentication if necessary.

account:

password:

retype password:

---

Enter the addresses of recipients to which mails with log files are automatically forwarded.  
Separate multiple recipients by commas

account from which mails are sent:

recipients of mails:

recipients of carbon copies:

recipients of blind carbon copies:

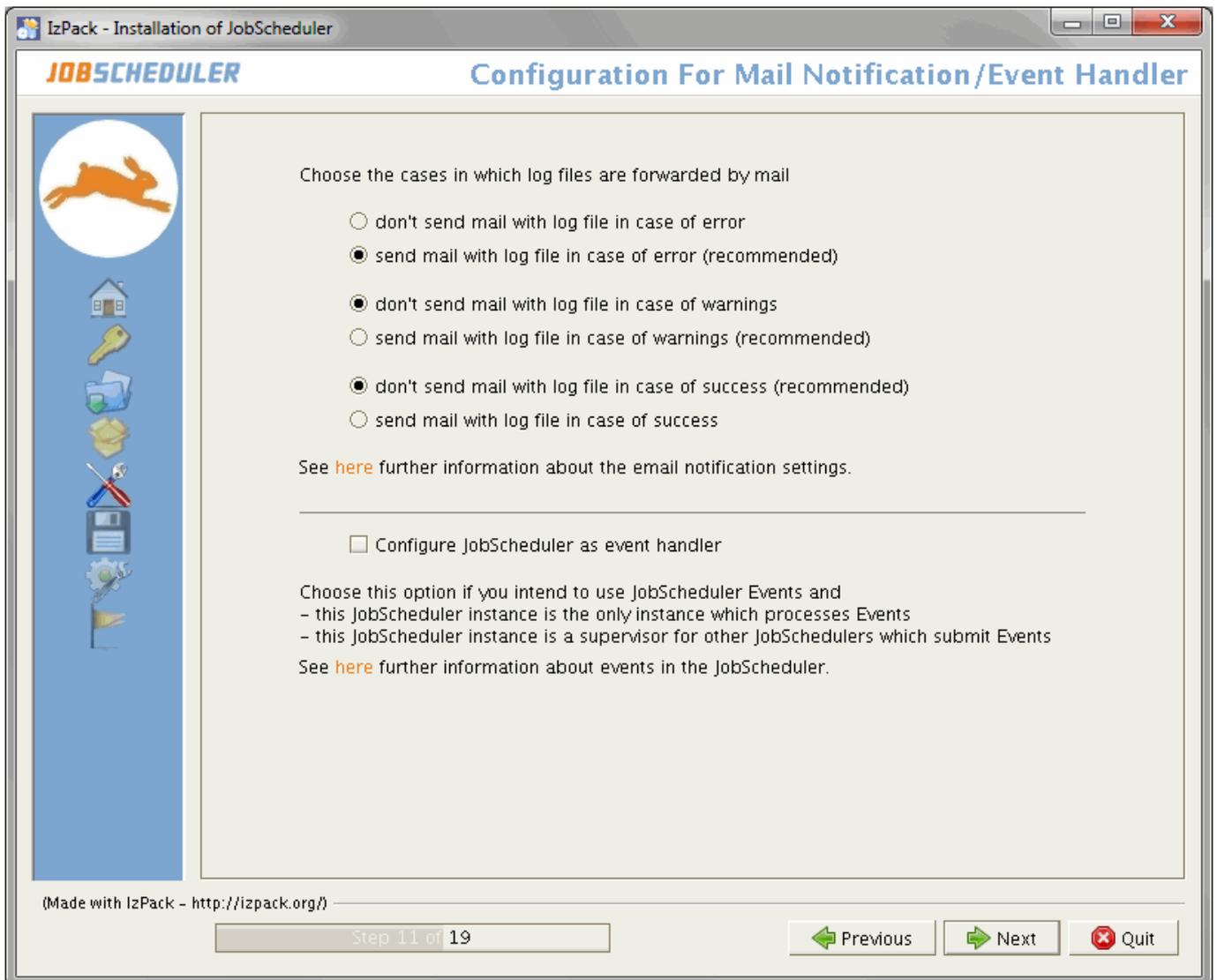
(Made with IzPack - <http://izpack.org/>)

Step 10 of 19

← Previous    Next →    ✖ Quit

Geben Sie hier Ihren SMTP-Server an und bestimmen Sie Absender, Empfänger und ggf. CC und BCC des Mailversands. Bei Angabe mehrere Email-Adressen müssen diese durch Komma getrennt werden.

Diese Angaben sind konfiguriert in der Datei `$SCHEDULER_DATA/config/factory.ini`, die nachträglich [manuell geändert](#) (Seite 37) werden kann.



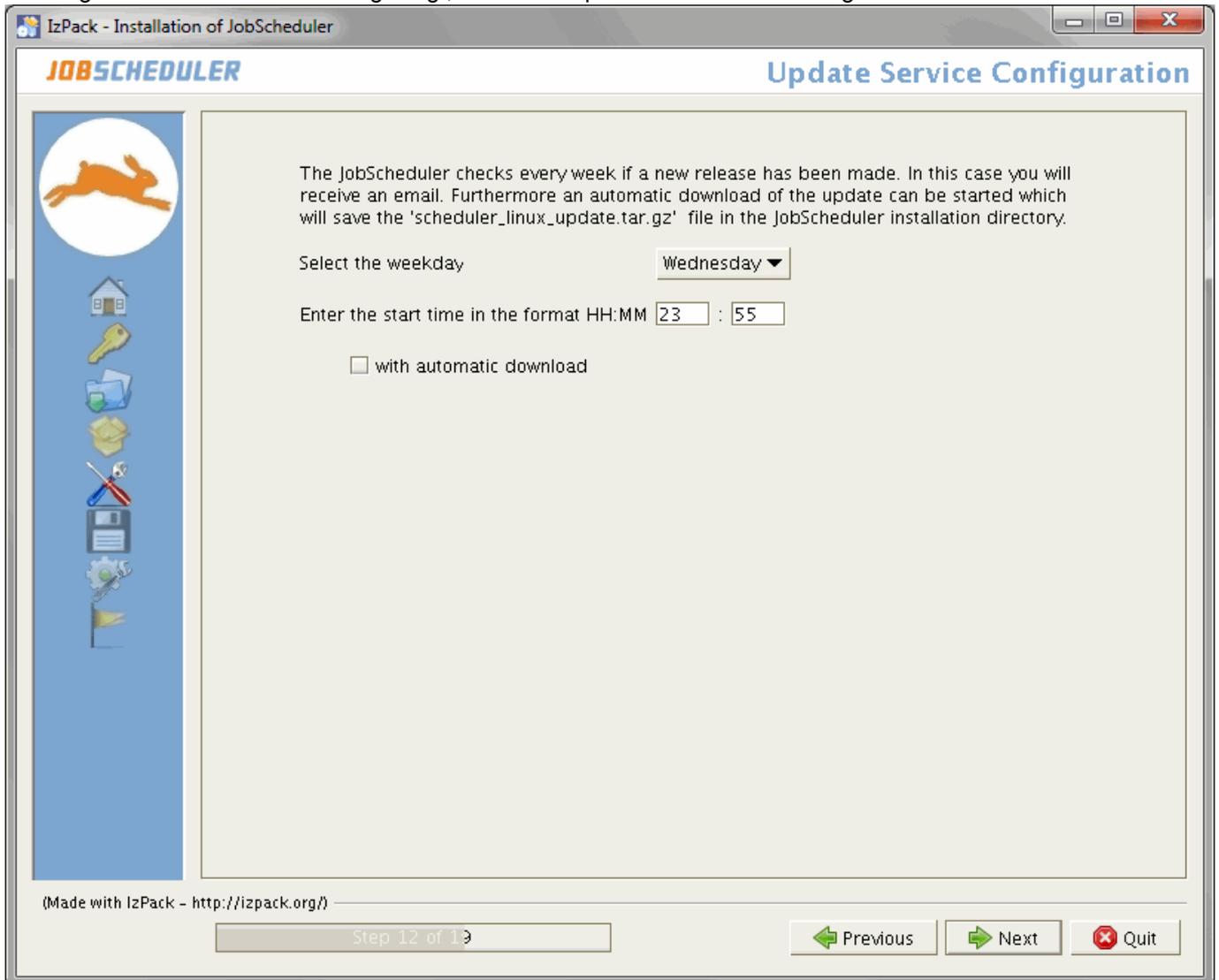
Hier bestimmen Sie, wann der **JobScheduler** Job-Protokolle automatisch versenden soll.

Diese Angaben sind konfiguriert in der Datei `$SCHEDULER_DATA/config/factory.ini`, die nachträglich [manuell geändert](#) (Seite 37) werden kann.

Der untere Teil wird nur angezeigt, wenn das Housekeeping-Paket gewählt wurde. Hier kann der **JobScheduler** als Event Handler konfiguriert werden. Entsprechende Objekte werden ggf. in `$SCHEDULER_DATA/config/live/sos/events` erzeugt. Näheres hierzu entnehmen Sie bitte der [Events Dokumentation](#).

## 1.6.2 Formular des Pakets Update Service

Das folgende Formular wird nur angezeigt, wenn das Update Service Paket ausgewählt wurde.



**Update Service Configuration**

The JobScheduler checks every week if a new release has been made. In this case you will receive an email. Furthermore an automatic download of the update can be started which will save the 'scheduler\_linux\_update.tar.gz' file in the JobScheduler installation directory.

Select the weekday Wednesday ▾

Enter the start time in the format HH:MM 23 : 55

with automatic download

(Made with IzPack - <http://izpack.org/>)

Step 12 of 13 
← Previous
Next →
✖ Quit

Dem **JobScheduler** wird ein *Job* hinzugefügt, der wöchentlich prüft, ob ein neueres Release vorliegt. Diesem *Job* können drei Parameter zugewiesen werden. Der Wochentag und die Uhrzeit an der der Job gestartet werden soll und ob ein automatischer Download erfolgt. Im Falle des automatischen Downloads wird je nach Betriebssystem eine Datei namens

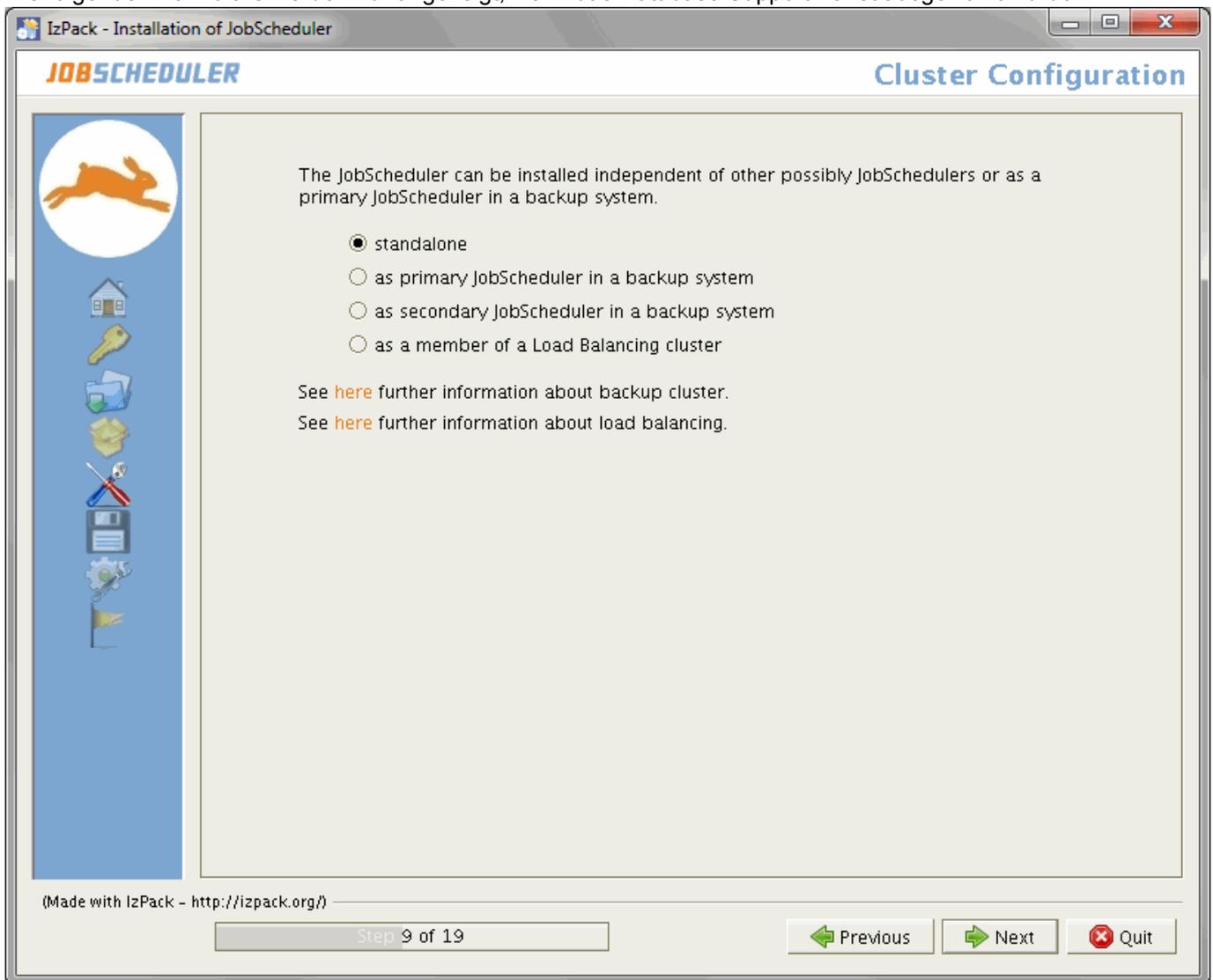
- scheduler\_win32\_update.zip
- scheduler\_linux32\_update.tar.gz
- scheduler\_solarisx86\_update.tar.gz
- scheduler\_solaris-sparc\_update.tar.gz
- scheduler\_hpux-ia64-32\_update.tar.gz
- scheduler\_aix\_update.tar.gz

in `$_SCHEDULER_DATA` gespeichert.

Für nachträgliche Konfiguration des Jobs öffnen Sie bitte die Datei `$SCHEDULER_DATA/config/live/sos/update/scheduler_check_updates.job.xml`. Weiteres zum Update-Verfahren entnehmen Sie bitte der [Update Service Dokumentation](#).

### 1.6.3 Formulare des Pakets Database Support

Die folgenden Formulare werden nur angezeigt, wenn das Database Support Paket ausgewählt wurde.



**JOBSCHEDULER** Cluster Configuration

The JobScheduler can be installed independent of other possibly JobSchedulers or as a primary JobScheduler in a backup system.

- standalone
- as primary JobScheduler in a backup system
- as secondary JobScheduler in a backup system
- as a member of a Load Balancing cluster

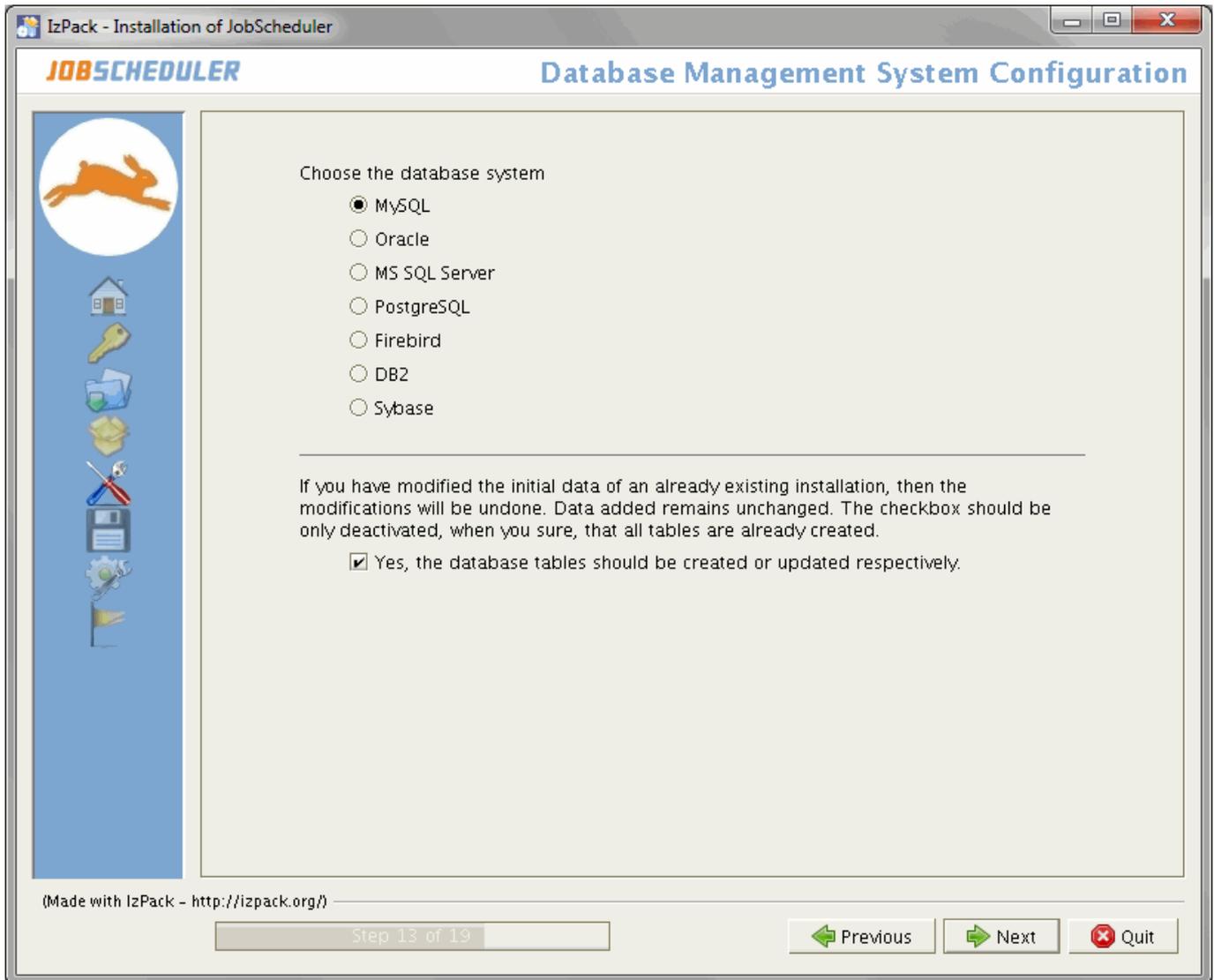
See [here](#) further information about backup cluster.  
See [here](#) further information about load balancing.

(Made with IzPack - <http://izpack.org/>)

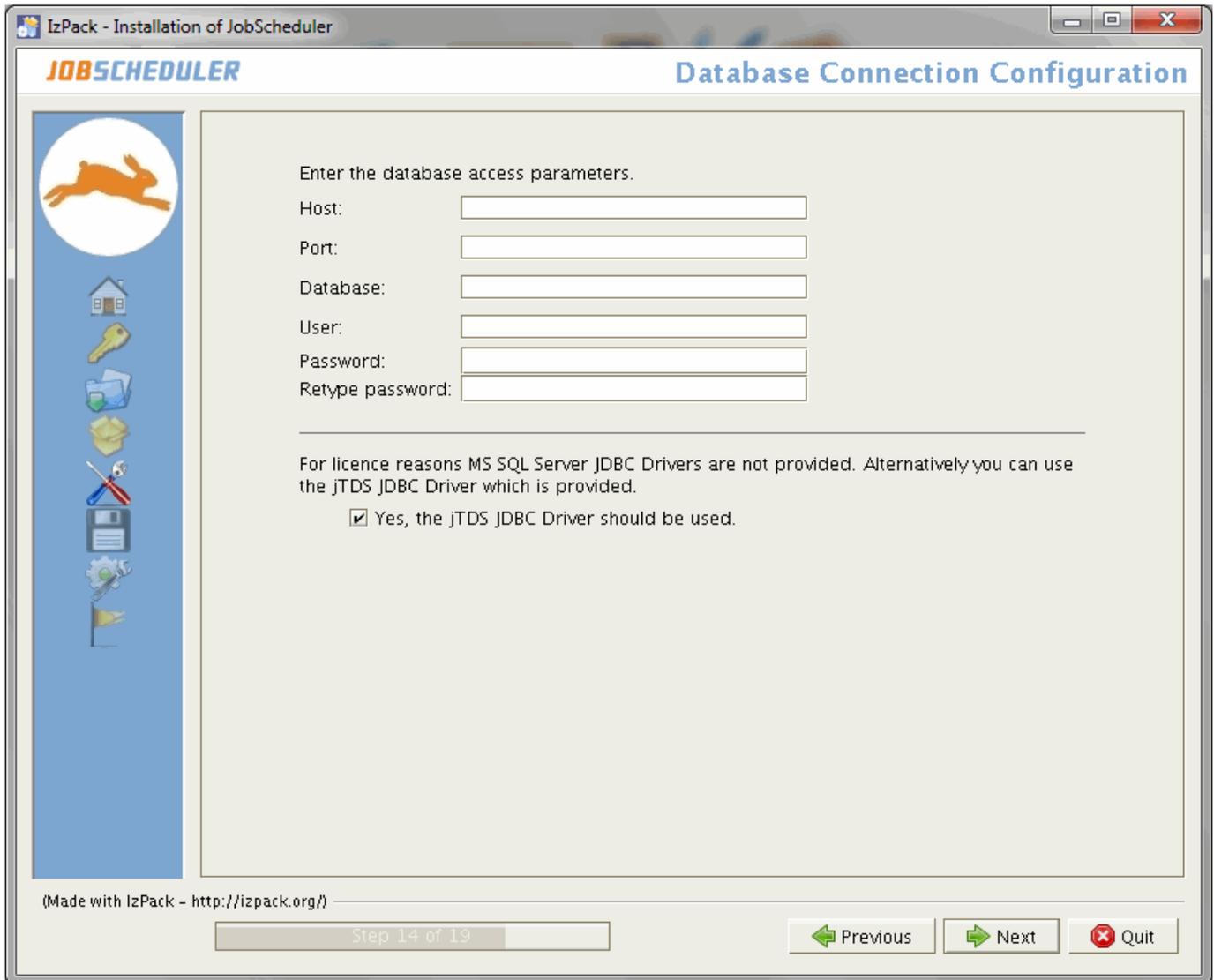
Step 9 of 19

Previous Next Quit

Mit obigen Radiobuttons bestimmen Sie, ob der **JobScheduler** "standalone" oder geclustert in einem Backup oder Load Balancing System installiert werden soll (siehe auch [Installation eines Clusters](#) (Seite 42)). Weitere Informationen über Backup Cluster finden Sie [hier](#), über Load Balancing [hier](#).



In obiger Auswahl bestimmen Sie Ihr Datenbanksystem. Ist die Checkbox in der Mitte angekreuzt (empfohlen), so ruft das Setup ein Skript auf, dass die nötigen Tabellen einrichtet und füllt. Andernfalls müssen Sie die [Tabellen manuell](#) (Seite 28) einrichten. Sollten Sie bereits einen anderen **JobScheduler** mit derselben Datenbank-Verbindung installiert haben, so müssen die Tabellen nicht erneut eingerichtet werden.



Hier ist die Datenbank-Verbindung anzugeben. Der mittlere Teil, in dem die Verwendung des mitgelieferten jTDS JDBC- gewählt werden kann, wird nur für Sybase und MS SQL Server angeboten. Habe Sie zuvor MySQL gewählt, dann wird Ihnen die alternativ Nutzung des mariadb JDBC Treiber angeboten. Wenn der jTDS JDBC-Treiber oder der mariadb JDBC Treiber ausgewählt wird, dann wird im Anschluss ein weiteres Formular Ihnen die Möglichkeit geben, Ihren eigenen JDBC-Treiber anzugeben.

Diese Angaben sind konfiguriert in den Dateien `$SCHEDULER_DATA/config/factory.ini`, `$SCHEDULER_DATA/config/hibernate.cfg.xml` and `$SCHEDULER_DATA/config/sos_settings.ini`. Alle Dateien können nachträglich [manuell geändert](#) (Seite 37) werden.



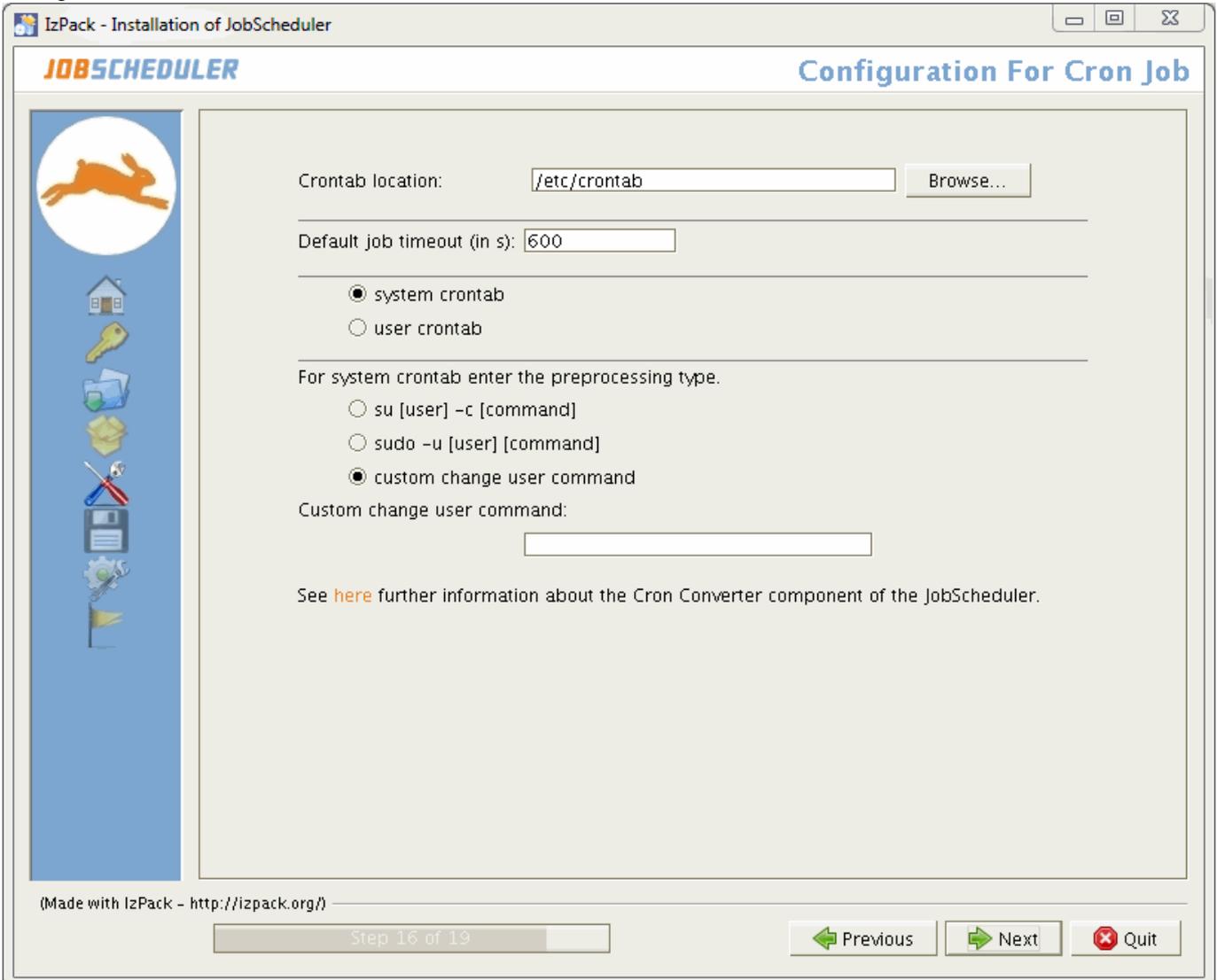
Dieses Formular erscheint nur, wenn Sie MySQL, MS SQL Server oder Sybase verwenden und zuvor auf den mariadb JDBC-Treiber für MySQL oder auf den jTDS JDBC-Treiber für MS SQL Server und Sybase verzichtet haben. Das Setup ruft später ein Skript zum Einrichten der Datenbank-Tabellen auf, das den JDBC-Treiber Ihres Datenbanksystems benötigt. Für Oracle, PostgreSQL, Firebird und DB2 sind diese im Setup enthalten, während aus lizenzrechtlichen Gründen Sie hier den MySQL-, MS SQL- bzw. Sybase-JDBC-Treiber angeben müssen, der zu Ihrer Datenbank-Version passt. Für MS SQL Server bzw. Sybase können Sie auch den mitgelieferten jTDS-JDBC-Treiber, für MySQL den mitgelieferten mariadb-JDBC-Treiber nutzen, in dem Sie die entsprechende Checkbox im vorherigen Dialog aktivieren.

Da im weiteren auch der **JobScheduler** einen JDBC-Treiber Ihres Datenbanksystems benötigt, wird dieser vom Setup nach `$SCHEDULER_HOME/lib` kopiert.

Ist das gewählte Datenbanksystem Firebird, dann muss sicher gestellt sein, dass während der Installation keine anderen Verbindungen zum Datenbank-Server existieren.

## 1.6.4 Formular des Pakets Cron Job

Das folgende Formular wird nur angezeigt, wenn das Cron Paket ausgewählt wurde. Dieser Dialog ist nur für Unix verfügbar.



**JOBSCHEDULER** Configuration For Cron Job

Crontab location:

Default job timeout (in s):

system crontab  
 user crontab

For system crontab enter the preprocessing type.

su [user] -c [command]  
 sudo -u [user] [command]  
 custom change user command

Custom change user command:

See [here](#) further information about the Cron Converter component of the JobScheduler.

(Made with IzPack - <http://izpack.org/>)

Step 16 of 19

Weitere Informationen über den Cron Converter finden Sie [hier](#).

## 2 Batch-Installation

Wird das Setup mit einer XML-Datei parametrisiert aufgerufen, so werden keine Formulare geöffnet. Im ausgepackte Verzeichnis `./jobscheduler.[release]` ist die Datei `scheduler_install.xml` enthalten. Diese Datei kann hierfür verwendet werden. Nachdem Sie in der Datei Installationspfade, Host, Port, Datenbank-Verbindung, usw. angegeben haben, wird das Setup, wie folgt, gestartet.

```
/tmp/jobscheduler.[release]> setup.sh scheduler_install.xml
```

**Beispiel: Installationsprogramm auf Unix mit Parameter-Datei starten**

```
C:\Windows\Temp\jobscheduler.[release]>setup.cmd scheduler_install.xml
```

**Beispiel: Installationsprogramm auf Windows mit Parameter-Datei starten**

Für das Setup sind unter Windows Administrator-Rechte erforderlich. Unter Windows öffnet das Setup gegebenenfalls einen entsprechenden Dialog. In Unix-Umgebungen wird ein sudo-Prompt geöffnet. Starten Sie das Setup unter Unix nicht als root, sondern benutzen Sie sudo!

Unter Unix kann auf die root-Rechte verzichtet werden. Wenn Sie den Job Scheduler ohne root-Rechte installieren möchten, dann rufen Sie

```
/tmp/jobscheduler.[release]> setup.sh -u scheduler_install.xml
```

**Beispiel: Installationsprogramm auf Unix mit Parameter-Datei starten, jedoch ohne root Rechte**

Die `scheduler_install.xml` hat den folgenden Inhalt:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!--
XML configuration file for JobScheduler setup

The JobScheduler is available with a dual licensing model.
- GNU GPL 2.0 License (see http://www.gnu.org/licenses/gpl-2.0.html)
- JobScheduler Commercial License (see licence.txt)

The setup asks you for the desired license model
(see <entry key="licenceOptions" .../> below).

If you call the setup with this XML file then you accept
at the same time the terms of the chosen license agreement.
-->
<AutomatedInstallation langpack="eng">
  <com.izforge.izpack.panels.UserInputPanel id="home">
    <userInput/>
  </com.izforge.izpack.panels.UserInputPanel>
  <com.izforge.izpack.panels.UserInputPanel id="licences">
    <userInput>

      <!-- Select the license model (GPL or Commercial) -->
      <entry key="licenceOptions" value="GPL"/>

      <!-- If you selected GPL as license model than the licence must be empty.
           Otherwise please enter a license key if available.
           It is also possible to modify the license key later. -->
      <entry key="licence" value=""/>

    </userInput>
  </com.izforge.izpack.panels.UserInputPanel>
  <com.izforge.izpack.panels.HTMLLicencePanel id="gpl_licence"/>
  <com.izforge.izpack.panels.HTMLLicencePanel id="commercial_licence"/>
  <com.izforge.izpack.panels.TargetPanel id="target">

    <!-- SELECT THE INSTALLATION PATH FOR THE BINARIES AND LIBRARIES
         The installation expands this path with the JobScheduler ID as subdirectory.
         The path must be absolute!
    -->
  </com.izforge.izpack.panels.TargetPanel>
</AutomatedInstallation>
```

```

Default paths are
/opt/sos-berlin.com/jobscheduler for Unix
C:\Program Files\sos-berlin.com\jobscheduler for Windows -->
<installpath><marked>[:choose absolute installation path of the JobScheduler:]</marked></installpath>

</com.izforge.izpack.panels.TargetPanel>
<com.izforge.izpack.panels.UserPathPanel id="userpath">

  <!-- SELECT THE DATA PATH FOR CONFIGURATION AND LOG FILES
  The installation expands this path with the JobScheduler ID as subdirectory.
  The path must be absolute!
  Default paths are
  /home/[user]/sos-berlin.com/jobscheduler for Unix
  C:\ProgramData\sos-berlin.com\jobscheduler for newer Windows
  C:\Documents and Settings\All Users\Application Data\sos-berlin.com\jobscheduler for older Windows -->
  <UserPathPanelElement>[:choose absolute data path of the JobScheduler configuration and log files:]<
/UsePathPanelElement>

</com.izforge.izpack.panels.UserPathPanel>
<com.izforge.izpack.panels.PacksPanel id="package">

  <!-- SELECT THE PACKS WHICH YOU WANT INSTALL -->

  <!-- Package: JobScheduler
  JobScheduler Basic Installation
  THIS PACK IS REQUIRED. IT MUST BE TRUE -->
  <pack index="0" name="Job Scheduler" selected="true"/>

  <!-- Package: Update Service
  It checks every week, if a new release has been made. -->
  <pack index="1" name="Update Service" selected="false"/>

  <!-- Package: Database Support
  Job history and log files can be stored in a database. Database support is
  available for MySQL, PostgreSQL, Firebird, Oracle, SQL Server, DB2.
  This package is strongly recommended. -->
  <pack index="2" name="Database Support" selected="true"/>

  <!-- Package: Housekeeping Jobs
  Housekeeping Jobs are automatically launched by the Job Scheduler, e.g. to send
  buffered logs by mail, to remove temporary files or to restart the JobScheduler. -->
  <pack index="5" name="Housekeeping Jobs" selected="true"/>

  <!-- Package: Cron Job
  THIS PACKAGE IS ONLY FOR UNIX.
  The Cron Adapter Job can be used to configure the JobScheduler with a crontab file.
  For that purpose, the Job reads the crontab file and dynamically adjusts the
  JobScheduler configuration. -->
  <pack index="7" name="Cron" selected="false"/>

</com.izforge.izpack.panels.PacksPanel>
<com.izforge.izpack.panels.UserInputPanel id="network">
  <userInput>
    <!-- Network Configuration -->

    <!-- Enter the name or ip address of the host on which the JobScheduler is operated -->
    <entry key="schedulerHost" value=""/>

    <!-- Enter the port for TCP communication -->
    <entry key="schedulerPort" value="4444"/>

    <!-- Enter the port for Jetty HTTP -->
    <entry key="jettyHTTPPort" value="44440"/>

    <!-- Enter the port for Jetty HTTPS -->
    <entry key="jettyHTTPSPort" value="8443"/>

    <!-- To enter a JobScheduler ID is required.
    The IDs of multiple instances of the JobScheduler must be unique per server.
    The JobScheduler ID expands the above installation paths as subdirectory.
    Please omit special characters like: / \ : ; * ? ! $ % & " < > ( ) | ^ -->
    <entry key="schedulerId" value="scheduler"/>

    <!-- It is recommended to enable TCP access for the host where the JobScheduler will install,
    optionally enter additional host names or ip addresses. To enable all hosts in your
    network to access the JobScheduler enter '0.0.0.0'. -->
    <entry key="schedulerAllowedHost" value="localhost"/>

    <!-- Choose (yes or no) whether the JobScheduler should be started at the end of the installation -->

```

```

    <entry key="launchScheduler" value="yes"/>

</userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel id="cluster">
  <userInput>
    <!-- Cluster Configuration -->

    <!-- The JobScheduler can be installed independent of other possibly JobSchedulers,
    as a primary JobScheduler in a backup system or as a backup JobScheduler.
    Use '' for a standalone, '-exclusive' for a primary
    or '-exclusive -backup' for a backup JobScheduler.
    A database is required for a backup system. All JobSchedulers in a backup system
    must have the same JobScheduler ID and the same database.
    Further you can set '-distributed-orders' for a load balancing cluster.
    For more information see
    http://www.sos-berlin.com/doc/en/scheduler/sos_help.htm?help_URL=scheduler.backup.htm
    http://www.sos-berlin.com/doc/en/scheduler/sos_help.htm?help_URL=scheduler.distributed_orders.htm -->
    <entry key="clusterOptions" value=""/>

  </userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel id="smtp">
  <userInput>
    <!-- Mail Recipients Configuration / SMTP Authentication -->

    <!-- Enter the ip address or host name and port (default: 25) of your SMTP server -->
    <entry key="mailServer" value=""/>
    <entry key="mailPort" value="25"/>

    <!-- Configure the SMTP authentication if necessary. -->
    <entry key="smtpAccount" value=""/>
    <entry key="smtpPass" value=""/>

    <!-- Enter the addresses of recipients to which mails with log files are automatically
    forwarded. Separate multiple recipients by commas -->

    <!-- Account from which mails are sent -->
    <entry key="mailFrom" value=""/>

    <!-- Recipients of mails -->
    <entry key="mailTo" value=""/>

    <!-- Recipients of carbon copies: -->
    <entry key="mailCc" value=""/>

    <!-- Recipients of blind carbon copies -->
    <entry key="mailBcc" value=""/>

  </userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel id="email">
  <userInput>
    <!-- Mail Configuration / Event Handler -->

    <!-- Choose in which cases mails with log files are automatically forwarded. -->
    <entry key="mailOnError" value="yes"/>
    <entry key="mailOnWarning" value="yes"/>
    <entry key="mailOnSuccess" value="no"/>

    <!-- The Housekeeping package is required for configure JobScheduler as event handler
    Choose this option if you intend to use JobScheduler Events and
    - this JobScheduler instance is the only instance which processes Events
    - this JobScheduler instance is a supervisor for other JobSchedulers which submit Events -->
    <entry key="jobEvents" value="off"/>

  </userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel id="update">
  <userInput>
    <!-- Update Configuration
    These entries are only necessary if the package 'Update Service' is chosen. -->

    <!-- The JobScheduler checks every week if a new release has been made. In this case
    you will receive an email. Furthermore an automatic download of the update can be
    started which will save the 'scheduler_(win32|linux|solaris)_update.(zip|tar.gz)'
    file in the JobScheduler installation directory. -->

    <!-- Enter the start time in the format HH:MM -->

```

```

<entry key="checkForUpdateStarttime" value="20:00"/>

<!-- Select the weekday via '0' for sunday, '1' for monday , ... and '6' for saturday. -->
<entry key="checkForUpdateStartday" value="1"/>

<!-- Choose '1' for automatic download, otherwise '0'. -->
<entry key="autoUpdateDownload" value="0"/>

</userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel id="database">
  <userInput>
    <!-- Database Configuration
    These entries are only necessary if the package 'Database Support' is chosen.-->

    <!-- Choose the database management system. Supported values are 'mysql' for MySQL,
    'oracle' for Oracle, 'mssql' for MS SQL Server, 'pgsql' for PostgreSQL,
    'fbsql' for Firebird, 'db2' for DB2 and 'sybase' for Sybase. -->
    <entry key="databaseDbms" value="mysql"/>

    <!-- You can choose between 'on' or 'off' to create the database tables.
    If you have modified the initial data of an already existing installation,
    then the modifications will be undone. Data added remains unchanged.
    This entry should be only 'off', when you sure, that all tables are already created. -->
    <entry key="databaseCreate" value="on"/>

  </userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel id="dbconnection">
  <userInput>
    <!-- Database Configuration
    These entries are only necessary if the package 'Database Support' is chosen. -->

    <!-- Enter the name or ip address of the database host -->
    <entry key="databaseHost" value=""/>

    <!-- Enter the port number for the database instance. Default ports are for MySQL 3306,
    Oracle 1521, MS SQL Server 1433, postgresSQL 5432, Firebird 3050, DB2 50000, Sybase 5000. -->
    <entry key="databasePort" value=""/>

    <!-- Enter the schema -->
    <entry key="databaseSchema" value=""/>

    <!-- Enter the user name for database access -->
    <entry key="databaseUser" value=""/>

    <!-- Enter the password for database access -->
    <entry key="databasePassword" value=""/>

    <!-- You must provide the MySQL, MS SQL Server or Sybase JDBC Driver respectively if you selected
    corresponding DBMS type. For license reasons MySQL, Sybase and MS SQL Server JDBC Drivers are
    not provided. Alternatively you can use the mariadb JDBC Driver for MySQL and
    the jTDS JDBC Driver for MS SQL Server and Sybase which is provided.-->

    <!-- You can choose between 'yes' or 'no' for using the jTDS JDBC Driver
    This entry has only an effect for MS SQL Server or Sybase -->
    <entry key="connectorJTDS" value="yes"/>

    <!-- You can choose between 'yes' or 'no' for using the mariadb JDBC Driver
    This entry has only an effect for MySQL -->
    <entry key="connectorMaria" value="yes"/>

  </userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel id="jdbc">
  <userInput>
    <!-- Configuration for JDBC Driver
    This entry is only necessary if the package 'Database Support' is chosen and you
    selected a DBMS type like MySQL, MS SQL Server or Sybase in the previous
    <userInput> element. -->

    <!-- You must provide the MySQL, MS SQL Server or Sybase JDBC Driver respectively if you selected
    corresponding DBMS type. For license reasons MySQL and MS SQL Server JDBC Drivers are
    not provided. Specify the JDBC Driver source (e.g. mysql-connector-java-*.jar for MySQL,
    sqljdbc.jar for MS SQL Server, jconn3.jar for Sybase). Alternatively you can use the mariadb
    JDBC Driver for MySQL and the jTDS JDBC Driver for MS SQL Server and Sybase which is provided. -->

    <!-- Select the path to JDBC Driver -->
    <entry key="connector" value=""/>
  </userInput>
</com.izforge.izpack.panels.UserInputPanel>

```

```

</userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel id="cron">
  <userInput>
    <!-- Configuration for Cron Job
         This input panel is ONLY FOR UNIX AVAILABLE
         These values will be ignored under Windows -->

    <!-- Enter the crontab path -->
    <entry key="cronCrontab" value="/etc/crontab"/>

    <!-- Select system crontab (1) or user crontab (0) -->
    <entry key="cronSystab" value="1"/>

    <!-- Enter the default job timeout (in s)
         The value must greater than 0 -->
    <entry key="cronTimeout" value="600"/>

    <!-- For system crontabs enter the preprocessing type
         su      for su [user] -c [command]
         sudo   for sudo -u [user] [command]
         (empty) for custom change user command -->
    <entry key="cronChangeUser" value=""/>

    <!-- Enter the custom change user command -->
    <entry key="cronChangeCommand" value=""/>

  </userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.InstallPanel id="install"/>
<com.izforge.izpack.panels.ProcessPanel id="process"/>
<com.izforge.izpack.panels.FinishPanel id="finish"/>
</AutomatedInstallation>

```

Beispiel: scheduler\_install.xml

Diese XML-Datei spiegelt alle Werte wider, die während eines Setup-Dialogs angegeben werden können.

## 3 Einrichten der Datenbank

Es wird empfohlen, dem **JobScheduler** einen Datenbankbenutzer mit dazugehöriger Datenbank (bzw. Schema) zur Verfügung zu stellen. Wie Datenbanken eingerichtet werden, entnehmen Sie bitte der Dokumentation Ihres Datenbanksystems. Unterstützt werden MySQL 5.x, Oracle (8.1.7, 9.2, 10g, 11g), Microsoft SQL Server (2000, 2005), PostgreSQL (8.x, 9.x), Firebird 1.5, DB2 8.x und Sybase ASE 15.0. Die Tabellen werden vom Setup erstellt, sofern während des Setups das Paket [Database Support](#) (Seite 11) ausgewählt und im entsprechenden Setup-Dialog die Datenbankverbindung angegeben wurde.

Das Setup benötigt zudem hinreichende Datenbank-Rechte für CREATE TABLE, INSERT, UPDATE, DELETE und SELECT Statements.

Des weiteren wird vom Setup die Datenbankverbindung in den folgenden [Konfigurationsdateien](#) (Seite 37) eingetragen.

- `$(SCHEDULER_DATA)/config/factory.ini` (genutzt von **JobScheduler**)
- `$(SCHEDULER_DATA)/config/hibernate.cfg.xml` (genutzt von **JID**)
- `$(SCHEDULER_DATA)/config/sos_settings.ini` (genutzt von `scheduler_install_tables.(sh|cmd)`)

### 3.1 MySQL

Aus lizenzrechtlichen Gründen ist kein MySQL JDBC Treiber im Setup enthalten. Bitte laden Sie sich diesen vor der **JobScheduler** Installationen herunter.

### 3.2 MS SQL Server and Sybase

Aus lizenzrechtlichen Gründen ist weder ein MS SQL noch ein Sybase JDBC Treiber im Setup enthalten. Alternativ können sie aber den im Setup enthaltenen JTDS JDBC Treiber verwenden. Anderenfalls laden Sie sich bitte einen MS SQL oder Sybase JDBC Treiber vor der **JobScheduler** Installationen herunter.

### 3.3 PostgreSQL

PostgreSQL benötigt PL/SQL. Prüfen Sie die verfügbaren Sprachen mit

```
createlang -U postgres -l scheduler
```

wobei "postgres" der Benutzername und "scheduler" der Datenbankname ist. "createlang" finden Sie im `bin`-Verzeichnis Ihrer PostgreSQL-Installation. Sollte "plpgsql" in der Ausgabe obigen Aufrufs nicht enthalten sein, dann fügen Sie bitte die Sprache hinzu durch

```
createlang -U postgres plpgsql scheduler
```

Es müssen folgende zwei PostgreSQL Server Variablen wie folgt gesetzt sein.

- `standard_conforming_strings = off`
- `bytea_output = 'escape'`

Sie brauchen diese Variablen nicht notwendigerweise global setzen. Es reicht auf User Ebene:

```
alter user scheduler set standard_conforming_strings = off;  
alter user scheduler set bytea_output = 'escape';
```

Beispiel: `standard_conforming_strings` und `bytea_output` für User setzen

wobei "scheduler" der User der **JobScheduler** Datenbank ist.

### 3.4 Manuelles Einrichten der Tabellen

Wurden die Tabellen nicht vom Setup bei der Installation des **JobScheduler** erstellt, so können Sie dieses manuell nachholen, in dem Sie das Skript `$SCHEDULER_HOME/install/scheduler_install_tables.(sh|cmd)` aufrufen.

Prüfen Sie, ob in der [Konfigurationsdatei](#) (Seite 37) `$SCHEDULER_DATA/config/sos_settings.ini` die Datenbankverbindung richtig eingetragen ist.

## 4 Verzeichnisstruktur nach der Installation

Die Inhalte der nachfolgend aufgeführten Verzeichnisse sind sowohl abhängig von der Wahl der Pakete während des Setups, als auch vom Betriebssystem. In diesen Fällen ist der Paketname und/oder das Betriebssystem nach dem Verzeichnis oder der Datei in Klammern angefügt. Ist der Paketname und/oder das Betriebssystem einem Verzeichnis angefügt, so gilt die Abhängigkeit für alle Dateien in diesem Verzeichnis.

Folgende Verzeichnisstruktur liegt unterhalb von `$SCHEDULER_HOME` vor:

- + `bin` (Windows)
  - `dashboard.cmd` Startskript für **JID**
  - `hostjava.dll` Programmbibliothek
  - `hostole.dll` Programmbibliothek
  - `jobeditor.cmd` Startskript für **JOE**
  - `jobscheduler.cmd` Startskript des **JobScheduler**
  - `jobscheduler_environment_variables.cmd` Skript zum Setzen der **JobScheduler** Umgebung
  - `jobscheduler_event.cmd` Skript für das Event Handling
  - `jobscheduler_client.pl` Perl-Skript (TCP/UDP-Client für den Versand von XML-Kommandos an einen **JobScheduler**)
  - `scheduler.exe` **JobScheduler** Engine
  - `scheduler.exe.local` Datei zur lokalen Verwendung von DLLs
  - `spidermonkey.dll` Programmbibliothek für JavaScript (Mozilla)
  
- + `bin` (Unix)
  - `dashboard.sh` Startskript für **JID**
  - `jobeditor.sh` Startskript für **JOE**
  - `jobscheduler.sh` Startskript des **JobScheduler**
  - `jobscheduler_environment_variables.sh` Skript zum Setzen der **JobScheduler** Umgebung
  - `jobscheduler_event.sh` Skript für das Event Handling
  - `jobscheduler_client.pl` Perl-Skript (TCP/UDP-Client für den Versand von XML-Kommandos an einen **JobScheduler**)
  - `scheduler` **JobScheduler** Engine
  - `scheduler_safe.sh` Überwachungs-Skript zum automatisierten Neustart des **JobScheduler**
  - `setuid` Programm zur Ausführung von Skripten im Kontext anderer Benutzer, siehe FAQ

- + **db** SQL-Dateien zur Erzeugung der Datenbank-Tabellen
  - + **mssql** MS SQL Server 2000, 2005
    - **scheduler.sql**
    - **scheduler\_events.sql**
    - **scheduler\_loganalyzer.sql**
    - **scheduler\_sanity.sql**
    - **scheduler\_sanity\_insert.sql**
    - **sosdailyschedule.sql**
    - **sosftphistory.sql**
  - + **mysql** MySQL 5.x
    - **\*.sql** (siehe Verzeichnis mssql)
    - + **procedures** (MySQL Maintenance Jobs)
      - **scheduler\_job\_procedure.sql**
    - **scheduler\_user\_jobs.sql** (MySQL Maintenance Jobs)
  - + **oracle** Oracle 8.1.7, 9.2, 10g
    - **\*.sql** (siehe Verzeichnis mssql)
  - + **fbsql** Firebird 1.5
    - **\*.sql** (siehe Verzeichnis mssql)
  - + **pgsql** PostgreSQL 8.x
    - **\*.sql** (siehe Verzeichnis mssql)
    - **hibernate\_sequence.sql**
  - + **db2** IBM DB2 8
    - **\*.sql** (siehe Verzeichnis mssql)
  - + **sybase** Sybase 15.x
    - **\*.sql** (siehe Verzeichnis mssql)
- + **doc** Dokumentationen
- + **install**
  - **scheduler\_install\_tables.sh** Skript zum manuellen Ausführen obiger SQL Dateien (Unix)
  - **scheduler\_install\_tables.cmd** Skript zum manuellen Ausführen obiger SQL Dateien (Windows)
- + **lib**
  - **\*.jar** Java-Archive (für Java-Jobs)
  - **scheduler.dll** für Java-Debugging (Windows)
  - **\*.so, \*.sl, \*.a** Bibliotheken (Unix)

- + `operations_gui` HTML- und Javascript-Dateien von **JOC**
- + `Uninstaller` Deinstallations-Programm
- + `user_bin`
  - `dashboard_environment_variables.cmd-example` Skript-Beispiel zum Setzen der **JID** Umgebung (Windows)
  - `jobeditor_environment_variables.cmd-example` Skript-Beispiel zum Setzen der **JOE** Umgebung (Windows)
  - `jobscheduler_environment_variables.cmd-example` Skript-Beispiel zum Setzen der **JobScheduler** Umgebung (Windows)
  - `dashboard_environment_variables.sh-example` Skript-Beispiel zum Setzen der **JID** Umgebung (Unix)
  - `jobeditor_environment_variables.sh-example` Skript-Beispiel zum Setzen der **JOE** Umgebung (Unix)
  - `jobscheduler_environment_variables.sh-example` Skript-Beispiel zum Setzen der **JobScheduler** Umgebung (Unix)

Folgende Verzeichnisstruktur liegt unterhalb von `$$SCHEDULER_DATA` vor:

- + `config`
  - + `cache` Konfigurations-Verzeichnis im Workload **JobScheduler** (Replikation des remote-Verzeichnisses im Supervisor **JobScheduler**)
  - + `events` (Housekeeping Jobs/Event Handling)
  - + `live` Lokales Konfiguration-Verzeichnis des **JobScheduler** (*Hot Folder*)
    - + `sos`
      - + `events` (Event Handler Jobs)
      - + `housekeeping` (Housekeeping Jobs)
      - + `mysql` (MySQL Maintenance Jobs)
      - + `update` (Update Service Job)
  - + `operations_gui`
    - `custom.js` Konfiguration-Datei für **JOC**
  - + `remote` Konfiguration-Verzeichnis für ein Workload **JobScheduler** im Supervisor **JobScheduler**
    - `factory.ini` **JobScheduler**-Konfigurationsdatei
    - `hibernate.cfg.xml` Datenbankverbindung für Hibernate-Klassen
    - `jetty.xml` Konfigurationsdatei für Jetty
    - `scheduler.xml` **JobScheduler**-Konfigurationsdatei
    - `scheduler.xsd` Schemadefinition für XML-Konfigurationsdateien
    - `scheduler_mail.xsl` Stylesheet zur Gestaltung von Protokoll-E-mails

- `sos.ini` Lizenzdatei
- `sos_settings.ini` Datenbank-Verbindung
- `web.xml` Konfigurationsdatei für Jetty
- + `jobs` Dokumentationen der JITL (**J**ob**S**cheduler Integrated Template Library)
- + `logs` Ablage der Log-Dateien

## 5 Starten und Stoppen des JobScheduler

### 5.1 JobScheduler Demon für Unix

Unter Unix wird der **JobScheduler** als Demon betrieben. Zum Starten und Stoppen verwenden Sie bitte das Skript:

```
$$SCHEDULER_HOME/bin/jobscheduler.sh start
```

Beispiel: JobScheduler auf Unix starten

```
$$SCHEDULER_HOME/bin/jobscheduler.sh stop
```

Beispiel: JobScheduler auf Unix beenden

Neben start und stop verarbeitet dieses Skript weitere Parameter, z.B. debug, restart, abort und kill.

Um den **JobScheduler** automatisch nach einem Reboot des Rechners zu starten, kopieren Sie bitte dieses Skript zusammen mit der Datei `$$SCHEDULER_HOME/bin/jobscheduler_environment_variables.sh` in das entsprechende startup/shutdown Verzeichnis, normalerweise ist dies `/etc/init.d`.

Wenn der **JobScheduler** nicht startet, dann sehen Sie für den Grund bitte in das `$$SCHEDULER_DATA/logs/scheduler.log`. Gegebenenfalls werden abhängige Bibliotheken vermisst. Das können Sie wie folgt prüfen.

```
$$SCHEDULER_HOME/bin/jobscheduler.sh ldd
```

### 5.2 JobScheduler Dienst für Windows

Unter Windows wird der **JobScheduler** als Dienst installiert. Sie finden den Dienst, in dem Sie in der Systemsteuerung die Management-Konsole für Dienste öffnen und nach dem Dienstnamen suchen, der mit "SOS **JobScheduler**" beginnt. Nach dem Setup hat der Dienst das lokale Systemkonto. Der Dienst ist konfiguriert für einen automatischen Start nach einem Reboot des Rechners.

Sie können den Dienst in der Dienste-Management-Konsole starten und stoppen oder mit

```
net.exe start sos_scheduler_[JobSchedulerId]
```

Beispiel: JobScheduler-Dienst auf Windows starten

```
net.exe start sos_scheduler_[JobSchedulerId]
```

Beispiel: JobScheduler-Dienst auf Windows beenden

...wobei `[JobSchedulerId]` die Id des **JobScheduler** ist.

Überdies können Sie den *Job Scheduler* auch von der Kommandozeile starten. Stellen Sie bitte zunächst sicher, dass der Dienst nicht gestartet ist und verwenden Sie das folgende Skript:

```
$SCHEDULER_HOME\bin\jobscheduler.cmd start
```

**Beispiel: JobScheduler auf Windows starten**

```
$SCHEDULER_HOME\bin\jobscheduler.cmd stop
```

**Beispiel: JobScheduler auf Windows beenden**

Neben start und stop verarbeitet dieses Skript weitere Parameter, z.B. debug, restart, abort und kill.

Wenn der **JobScheduler** nicht startet, dann sehen Sie für den Grund bitte in das [\\$SCHEDULER\\_DATA/logs/scheduler.log](#).

## 6 JOC, JOE und JID öffnen

### 6.1 JOC (JobScheduler Operations Center) öffnen

**JOC (JobScheduler Operations Center)** ist eine Oberfläche zur Überwachung und Steuerung des **JobScheduler**. **JOC** wird im Browser (Internet Explorer und Firefox werden unterstützt) mit der folgenden URL geöffnet.

```
http://[scheduler_host]:[scheduler_port]/
```

**JOC** kann auch mit Jetty geöffnet werden. Nutzen Sie hierzu die folgende URLs.

```
http://[scheduler_host]:[jetty_http_port]/
```

```
https://[scheduler_host]:[jetty_https_port]/
```

Wenn Sie **JOC** von einem anderen Rechner aus öffnen, müssen Sie sicherstellen, dass die Kommunikation weder durch eine Firewall noch durch die [Security Einstellungen](#) des **JobScheduler** behindert wird.

Wenn Sie die **JobScheduler** Installation aktualisiert haben, kann es nötig sein, den Cache des Browsers zu löschen, damit Änderungen in **JOC** wirksam werden.

### 6.2 JOE (JobScheduler Object Editor) öffnen

**JOE (JobScheduler Object Editor)** ist eine Anwendung für die Erzeugung und Konfiguration der **JobScheduler** Objekte (*Jobs*, *Job-Ketten*, *Aufträge* and *Schedules*).

**JOE** wird gestartet mit ...

```
$SCHEDULER_HOME\bin\jobeditor.cmd
```

Beispiel: JOE auf Windows starten

```
$SCHEDULER_HOME/bin/jobeditor.sh
```

Beispiel: JOE auf Unix starten

Ein X-Server und GTK2 ist auf Unix Systemen erforderlich. Die hierfür erforderlichen Bibliotheken müssen die gleiche Bittigkeit haben wie **JOE**.

Wenn Sie **JOE** auf Windows starten, kann es sein, dass nicht passiert. Dann liegt eine initialer Fehler vor. Verwenden Sie debug als Argument des Aufrufs, um den Fehler zu sehen.

```
$SCHEDULER_HOME\bin\jobeditor.cmd debug
```

Beispiel: JOE auf Windows debuggen

Sollten Sie z.B. nach dem Start von **JOE** den Fehler

```
Cannot load 32-bit SWT libraries on 64-bit JVM
```

erhalten, dann müssen Sie die Umgebungsvariable `$JAVA_HOME` in `$SCHEDULER_HOME/user_bin/jobeditor_environment_variables.(sh|cmd)` (Seite 38) anpassen.

## 6.3 JID (JobScheduler Information Dashboard) öffnen

**JID (JobScheduler Information Dashboard)** ist eine Anwendung für eine Übersicht über die geplanten und tatsächlich gelaufenen *Jobs*. Weitere Informationen finden Sie in der [Dashboard Dokumentation](#)

**JID** wird gestartet mit ...

```
$SCHEDULER_HOME\bin\dashboard.cmd
```

Beispiel: JID auf Windows starten

```
$SCHEDULER_HOME/bin/dashboard.sh
```

Beispiel: JID auf Unix starten

Ein X-Server und GTK2 ist auf Unix Systemen erforderlich. Die hierfür erforderlichen Bibliotheken müssen üssen die gleiche Bittigkeit haben wie **JID**.

Wenn Sie **JID** auf Windows starten, kann es sein, dass nicht passiert. Dann liegt eine initialer Fehler vor. Verwenden Sie `debug` als Argument des Aufrufs, um den Fehler zu sehen.

```
$SCHEDULER_HOME\bin\jobeditor.cmd debug
```

Beispiel: JOE auf Windows debuggen

Sollten Sie z.B. nach dem Start von **JID** den Fehler

```
Cannot load 32-bit SWT libraries on 64-bit JVM
```

erhalten, dann müssen Sie die Umgebungsvariable `$JAVA_HOME` in [\\$SCHEDULER\\_HOME/user\\_bin/dashboard\\_environment\\_variables.\(sh|cmd\)](#) (Seite 38) anpassen.

## 7 Konfiguration

Die Konfiguration des **JobScheduler** erfolgt über folgende Dateien:

- `$$SCHEDULER_DATA/config/sos.ini`
- `$$SCHEDULER_DATA/config/factory.ini`
- `$$SCHEDULER_DATA/config/scheduler.xml`
- `$$SCHEDULER_DATA/config/operations_gui/custom.js` (für **JOC**)
- `$$SCHEDULER_DATA/config/hibernate.cfg.xml` (Datenbank-Verbindung von **JID**)
- `$$SCHEDULER_HOME/bin/jobscheduler_environment_variables.(sh|cmd)`

Diese werden vom Setup entsprechend Ihrer Angaben konfiguriert. Darüberhinaus können Sie weitere drei Dateien anlegt werden, um die Umgebung anzupassen. Diese sind:

- `$$SCHEDULER_HOME/user_bin/jobscheduler_environment_variables.(sh|cmd)` (für **JobScheduler**)
- `$$SCHEDULER_HOME/user_bin/jobeditor_environment_variables.(sh|cmd)` (für **JOE**)
- `$$SCHEDULER_HOME/user_bin/dashboard_environment_variables.(sh|cmd)` (für **JID**)

### 7.1 Die Datei sos.ini

In der Datei `$$SCHEDULER_DATA/config/sos.ini` ist der Lizenzschlüssel enthalten. Zudem können Java Optionen und wenn nötig der Ort der JVM gesetzt werden. Näheres hierzu ist [hier](#) beschrieben.

### 7.2 Die Datei factory.ini

In der Datei `$$SCHEDULER_DATA/config/factory.ini` werden insbesondere Email-Einstellungen, Datenbankverbindung und Klassenpfad der Java-Archive gesetzt. Näheres hierzu finden Sie [hier](#).

### 7.3 Die Datei scheduler.xml

In der Datei `$$SCHEDULER_DATA/config/scheduler.xml` wird u.a. der Port des **JobScheduler** bestimmt. Näheres hierzu ist [hier](#) beschrieben.

### 7.4 Die Datei custom.js

Die Datei `$$SCHEDULER_DATA/config/operations_gui/custom.js` wird von **JOC (JobScheduler Operations Center)** genutzt. In ihr kann neben anderen Einstellungen insbesondere die Sprache und Filter konfiguriert werden. Siehe auch [hier](#) für weitere Informationen.

## 7.5 Die Datei hibernate.cfg.xml

Die Datei `$$SCHEDULER_DATA/config/hibernate.cfg.xml` wird von **JID (JobScheduler Information Dashboard)** genutzt, um sich mit der Datenbank zu verbinden. Weitere Informationen finden Sie in der [Dashboard Dokumentation](#).

## 7.6 Die Datei jobscheduler\_environment\_variables.(sh|cmd)

In dieser Datei sind die Startparameter und die ID des **JobScheduler** Id konfiguriert

Diese Datei sollte nicht verändert werden, da die Änderungen nach einem **JobScheduler** Update verloren gegangen sein könnten. Wenn Umgebungsvariablen angepasst werden müssen, dann legen sie bitte eine Datei `$$SCHEDULER_HOME/user_bin/jobscheduler_environment_variables.(sh|cmd)` an. Verwenden Sie hierfür die Datei `$$SCHEDULER_HOME/user_bin/jobscheduler_environment_variables.(sh|cmd)-example` als Vorlage. Unter Unix muss die `$$SCHEDULER_HOME/user_bin/jobscheduler_environment_variables.(sh|cmd)` ausführbar sein.

Die Datei `$$SCHEDULER_HOME/bin/jobscheduler_environment_variables.sh` ist insbesondere für Unix relevant, da in ihr der `LD_LIBRARY_PATH` gesetzt wird. Wenn der **JobScheduler** die Java Virtual Machine nicht findet, dann muss der `LD_LIBRARY_PATH` oder `JAVA_HOME` entsprechend angepasst werden.

In diesem Fall wird folgender Fehler in `$$SCHEDULER_DATA/logs/scheduler.log` protokolliert:

```
[ERROR Z-JAVA-100 Java Virtual Machine cannot be loaded [0509-022 Cannot load module
... System error: A file or directory in the path name does not exist.] [libjvm.so]]
```

Bei Änderungen der **JobScheduler** Id muss unter Windows der entsprechende Dienst neu installiert werden.

```
$$SCHEDULER_HOME/bin/jobscheduler.cmd remove
$$SCHEDULER_HOME/bin/jobscheduler.cmd install
```

Beispiel: JobScheduler Dienst neu installieren.

## 7.7 Die Datei jobeditor\_environment\_variables.(sh|cmd)

In dieser Datei können die Startparameter von **JOE (JobScheduler Object Editor)** konfiguriert werden.

**JOE** wird mit dem Skript `$$SCHEDULER_HOME/bin/jobeditor.(sh|cmd)` gestartet. Diese Datei sollte nicht verändert werden, da die Änderungen nach einem **JobScheduler** Update verloren gehen sein könnten. Wenn Umgebungsvariablen (z.B. `$$JAVA_HOME`) angepasst werden müssen, dann legen sie bitte eine Datei `$$SCHEDULER_HOME/user_bin/jobeditor_environment_variables.(sh|cmd)` an. Verwenden Sie hierfür die Datei `$$SCHEDULER_HOME/user_bin/jobeditor_environment_variables.(sh|cmd)-example` als Vorlage. Unter Unix muss die `$$SCHEDULER_HOME/user_bin/jobeditor_environment_variables.(sh|cmd)` ausführbar sein.

## 7.8 Die Datei dashboard\_environment\_variables.(sh|cmd)

In dieser Datei können die Startparameter von **JID (JobScheduler Information Dashboard)** konfiguriert werden.

**JID** wird mit dem Skript `$$SCHEDULER_HOME/bin/dashboard.(sh|cmd)` gestartet. Diese Datei sollte nicht verändert werden, da die Änderungen nach einem **JobScheduler** Update verloren gehen könnten. Wenn

Umgebungsvariablen (z.B. `$JAVA_HOME`) angepasst werden müssen, dann legen sie bitte eine Datei `$$SCHEDULER_HOME/user_bin/dashboard_environment_variables.(sh|cmd)` an. Verwenden Sie hierfür die Datei `$$SCHEDULER_HOME/user_bin/dashboard_environment_variables.(sh|cmd)-example` als Vorlage. Unter Unix muss die `$$SCHEDULER_HOME/user_bin/dashboard_environment_variables.(sh|cmd)` ausführbar sein.

## 8 JobScheduler aktualisieren

Zur Aktualisierung des **JobScheduler** wird ein spezielles "Update-Setup" angeboten.

- `jobscheduler_linux_update-x64.[release].tar.gz` für Linux 64 bit
- `jobscheduler_linux_update-x86.[release].tar.gz` für Linux 32 bit
- `jobscheduler_windows_update-x64.[release].zip` für Windows 64 Bit
- `jobscheduler_windows_update-x86.[release].zip` für Windows 32 Bit
- `jobscheduler_solarisx86_update.[release].tar.gz` für Solaris
- `jobscheduler_solaris-sparc_update.[release].tar.gz` für Solaris
- `jobscheduler_hpux-ia64-32_update.[release].tar.gz` für HP-UX Itanium
- `jobscheduler_aix32_update.[release].tar.gz` für AIX

Aufruf dieses Setup läuft analog wie beim "[Full-Setup](#) (Seite 7)".

Sie können auch das "Full-Setup" zum Aktualisieren nutzen, jedoch ist das nicht vorbehaltlos geeignet. Wenn Sie eine **JobScheduler** Installation aktualisieren, deren Erstinstallation vor Release 1.3.9 erfolgte, so ist der Verzeichnisbaum ein anderer. Verwenden Sie hierfür "Update-Setup".

## 9 Mehrfache Installationen

Sie können beliebig viele **JobScheduler** installieren.

Folgendes ist im Formular [Netzwerk-Konfiguration](#) (Seite 13) des Basis-Pakets **JobScheduler** unbedingt zu beachten

- Die **JobScheduler ID** ist über alle installierten **JobScheduler** eindeutig zu wählen. Es sei denn, sie wollen einen [Cluster](#) (Seite 42) installieren.

Unter Windows bestimmt die ID des **JobScheduler** zudem den Namen des nach dem Setup installierten Dienstes in der Form `sos_scheduler_[scheduler_id]`. Dienstnamen müssen pro Rechner eindeutig sein.

- Der *TCP-Port* ist über alle installierten **JobScheduler** eines Rechners eindeutig zu wählen.

Es wird empfohlen allen installierten **JobScheduler** die gleiche Datenbank-Verbindung zuzuweisen. Insbesondere bei Verwendung eines Backup-Clusters oder dem Load Balancing ist dies zwingend.

## 10 Installation eines Clusters

Alle **JobScheduler** eines Clusters haben fast identische Setup-Konfigurationen. Insbesondere haben sie dieselbe **JobScheduler** Id und dieselbe Datenbank. Nach der Installation des ersten **JobScheduler** des Clusters ist die Datei `$$SCHEDULER_HOME/scheduler_install.xml` erzeugt worden. Diese Datei kann für eine [Batch-Installation](#) (Seite 22) der anderen geclusterten **JobScheduler** genutzt werden. Man braucht zuvor lediglich einige Werte in der `scheduler_install.xml` anpassen. Diese sind `schedulerHost` und bei einem Backup Cluster zusätzlich `clusterOptions`. Des weiteren setzen Sie den Wert von `databaseCreate` auf `off`, da bereits bei der Installation der primären **JobScheduler** die Datenbank eingerichtet wurde.

```
...
<entry key="schedulerHost" value="[other host]"/>
...
<!-- for Backup Cluster -->
<entry key="clusterOptions" value="-exclusive -backup"/>
...
<entry key="databaseCreate" value="off"/>
...
```

Beispiel: Schipsel von `$$SCHEDULER_HOME/scheduler_install.xml`

Weitere Informationen über Backup Cluster finden Sie [hier](#), über Load Balancing [hier](#).

## 11 Deinstallation

### 11.1 Deinstallation per Uninstaller

Der für die Installation ausgeführte Setup installiert auch das Deinstallations-Programm `$SCHEDULER_HOME/Uninstaller/uninstall.jar`. Gestartet wird die Deinstallation mittels

```
unix-shell> $SCHEDULER_HOME/Uninstaller/uninstall.sh
```

Beispiel: Deinstallationprogramm auf Unix starten

```
windows-shell> $SCHEDULER_HOME\Uninstaller\uninstall.cmd
```

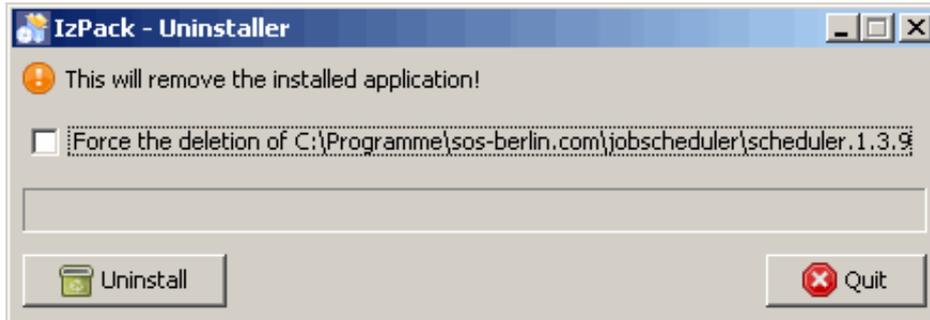
Beispiel: Deinstallationprogramm auf Windows starten

Auf Unix fordert das Deinstallationsprogramm root Rechte mittels `sudo`. Wenn der **JobScheduler** ohne root Rechte installiert wurde, dann benötigen Sie auch keine bei der Deinstallation. In diesem Fall rufen sie das Skript wie folgt auf:

```
unix-shell> $SCHEDULER_HOME/Uninstaller/uninstall.sh -u
```

Beispiel: Deinstallationprogramm auf Unix ohne root Rechte starten

Auf Windows fordert anschließend ein Dialog zur Bestätigung der Deinstallation auf.



Eine ggf. eingerichtete Datenbank für den **JobScheduler** müssen Sie manuell löschen.

Der installierte Dienst "SOS **JobScheduler** id=[scheduler\_id]" auf Windows sollte nach der Deinstallation entfernt sein. Hierbei ist [scheduler\_id], die von Ihnen bei der Installation gewählte Id des **JobScheduler**. Mitunter ist dieser aber nur als deaktiviert gekennzeichnet. Dann ist der Dienst erst nach einem Neustart des Rechners entfernt. Sie können dies prüfen, in dem Sie den Dienste-Dialog öffnen (Start->Ausführen services.msc) oder die Kommandozeile

Sollte der Dienst nur deaktiviert sein, so ist eine erneute Installation eines **JobScheduler** mit gleicher [scheduler\_id] erst nach einem Neustart möglich.

### 11.2 Manuelle Deinstallation unter Windows

Für die manuelle Deinstallation öffnen Sie eine Kommandozeile (Start->Ausführen cmd) und führen dann folgende Schritte aus.

- **JobScheduler** beenden

```
$SCHEDULER_HOME\bin\jobscheduler.cmd stop
```

Wenn der **JobScheduler** bereits beendet war, wird eine Fehlermeldung erscheinen, die ignoriert werden kann.

- Dienst des **JobScheduler** deinstallieren

```
$SCHEDULER_HOME\bin\jobscheduler.cmd remove
```

- Datenbank entfernen

Zum Entfernen der für den **JobScheduler** eingerichteten Datenbank lesen Sie bitte die Dokumentation ihres Datenbanksystems.

- Programmbibliothek hostole.dll deregistrieren

```
regsvr32 /u $SCHEDULER_HOME\bin\hostole.dll
```

- Alle Dateien und Verzeichnisse löschen

```
rmdir /S /Q $SCHEDULER_HOME  
rmdir /S /Q $SCHEDULER_DATA
```

## 11.3 Manuelle Deinstallation unter Unix

Für die manuelle Deinstallation öffnen Sie eine Shell und führen dann folgende Schritte aus.

- **JobScheduler** beenden

```
$SCHEDULER_HOME/bin/jobscheduler.sh stop
```

Wenn der **JobScheduler** bereits beendet war, wird eine Fehlermeldung erscheinen, die ignoriert werden kann.

- Datenbank entfernen

Zum Entfernen der für den **JobScheduler** eingerichteten Datenbank lesen Sie bitte die Dokumentation ihres Datenbanksystems.

- Alle Dateien und Verzeichnisse löschen

```
rm -r -f $SCHEDULER_HOME  
rm -r -f $SCHEDULER_DATA
```

## 12 Verfahren des automatischen Updates

Auf <http://www.sos-berlin.com> ist ein Webservice installiert, der Anfragen zur aktuellen verfügbaren Version (Release) des **JobScheduler** entgegennimmt. Liegt eine neuere Version vor, wird diese Information dem Aufrufer zurückgeliefert.

Sofern im Setup das Paket Update Service gewählt wurde, ist ein **Job** ausgeliefert, der einmal wöchentlich eine Anfrage an den Webservice bezüglich neuerer Versionen macht. Liegt eine neue Version vor, wird eine entsprechende Email an den Systemadministrator versendet. Wahlweise kann der **Job** automatisch die erforderlichen Dateien herunterladen.

Näheres erfahren Sie in der [Update Service Dokumentation](#).

## 13 Betrieb des 32bit JobScheduler auf 64bit Systemen

Der 32-Bit **JobScheduler** kann auch auf den unterstützten Betriebssystemen mit 64bit Betrieb eingesetzt werden. Der 32-Bit **JobScheduler** benötigt weiterhin ein 32-Bit JRE und einige 32-Bit Bibliotheken (siehe [Voraussetzungen](#) (Seite 5)).

Sollte auch ein 64-Bit JRE installiert sein, dann kann es sein, dass das **JobScheduler** Setup das 64-Bit JRE verwendet und in die Datei ``${SCHEDULER_HOME}/bin/jobscheduler_environment_variables.(sh|cmd)` den Installationspfad des 64-Bit JRE in die Variable ``${JAVA_HOME}` schreibt. In diesem Fall wird der **JobScheduler** nicht starten und Sie müssen die ``${JAVA_HOME}` variable in ``${SCHEDULER_HOME}/user_bin/jobscheduler_environment_variables.(sh|cmd)` (Seite 38) ändern.

Wenn der **JobScheduler** nicht startet, dann sehen Sie für den Grund bitte in das ``${SCHEDULER_DATA}/logs/scheduler.log`. Gegebenenfalls werden abhängige Bibliotheken vermisst. Das können Sie wie folgt prüfen.

```
`${SCHEDULER_HOME}/bin/jobscheduler.sh ldd
```

### 13.1 32bit JOE and JID auf 64bit Systemen

Es kann passieren, dass weder **JOE** noch **JID** gestartet werden können, da ein 64-Bit JRE aufgerufen wird. In diesem Fall müssen Sie die ``${JAVA_HOME}` variable in ``${SCHEDULER_HOME}/user_bin/jobeditor_environment_variables.(sh|cmd)` (Seite 38) und ``${SCHEDULER_HOME}/user_bin/dashboard_environment_variables.(sh|cmd)` (Seite 38) ändern.

Für Linux and Windows gibt es eine 64-Bit version von **JOE** und **JID**. 32-Bit **JOE** and **JID** kann aber auch mit einem 64-Bit JRE gestartet werden. Folgende Schritte sind hierfür nötig.

- Laden Sie ein 64-Bit [swt.jar](#) herunter und kopieren es nach ``${SCHEDULER_HOME}/lib`.
- Set ``${JAVA_HOME}=[/path/to/64-Bit JRE]` in ``${SCHEDULER_HOME}/user_bin/jobeditor_environment_variables.(sh|cmd)` (Seite 38) and ``${SCHEDULER_HOME}/user_bin/dashboard_environment_variables.(sh|cmd)` (Seite 38).
- Unter Unix müssen die folgenden Bibliotheken dann in 64-Bit vorliegen.
  - `libgtk-x11-2.0.so.0`
  - `libXtst.so.6`

## 14 Troubleshooting

Sie finden Hinweise zur Problembhebung auf unserer [Home Page](#), in unseren [FAQ](#) oder im [Sourceforge Forum](#).

## Glossar

### Auftrag

Ein Auftrag aktiviert die Verarbeitung einer Job-Kette. Der Auftrag enthält die Parameter für die Jobs einer Job-Kette. Jeder Job der Job-Kette hat Zugriff auf die Parameter eines Auftrags. Auftrag-Parameter überschreiben gleichnamige Job-Parameter. Ein Auftrag kann eine Zeitsteuerung enthalten.

Ein Auftrag durchläuft nacheinander die Jobs einer Job-Kette. Tritt ein Verarbeitungsfehler in einem Job auf, dann kann konfiguriert werden, dass der Auftrag...

- aus der Job-Kette entfernt wird.
- von einem weiteren Job der Kette ausgeführt wird.
- vom fehlerhaften Job wiederholt ausgeführt wird.
- in der Kette am fehlerhaften Job stehen bleibt, das heißt die Verarbeitung des Auftrags wird ausgesetzt, bis dieser wieder manuell fortgesetzt wird.

### Job

Programme und Skripte, die vom **JobScheduler** ausgeführt werden sollen, müssen in Jobs eingebettet werden. Jobs können beliebige ausführbare Dateien starten oder Job-Skripte enthalten, die die Programmschnittstelle des **JobScheduler** verwenden. Jobs können in mehreren Instanzen (Tasks) ablaufen, wenn dies zur Skalierung der Leistung gewünscht ist.

Es wird zwischen Standalone-Jobs und Auftrags-Jobs unterschieden. Während Auftrags-Jobs durch Aufträge innerhalb einer Job-Kette gestartet werden, können Standalone-Jobs davon unabhängig gestartet werden, das heißt manuell, durch eine Zeitsteuerung oder per Verzeichnis-Überwachung. Standalone-Jobs können nicht in Job-Ketten eingereiht werden.

### Job-Kette

Eine Reihe von Jobs, die nacheinander Aufträge verarbeiten. Der **JobScheduler** startet die Jobs einer Job-Kette automatisch, wenn ein Auftrag eintrifft. Job-Ketten bieten die Möglichkeit mehrere Aufträge parallel zu verarbeiten, indem die Jobs in mehreren Instanzen (Tasks) gestartet werden.

### JOC (JobScheduler Operations Center)

**JOC (JobScheduler Operations Center)** ist die Oberfläche des **JobScheduler** zur Überwachung und Steuerung der **JobScheduler** Objekte, also zum Beispiel Jobs, Job-Ketten und Aufträge.

**\_JOC** öffnen Sie im Browser mit `http://[scheduler_host]:[scheduler_port]`, wobei `[scheduler_host]` und `[scheduler_port]` der Hostname und die Nummer des TCP Ports des **JobScheduler** sind (z.B. `http://localhost:4444`).

### JOE (JobScheduler Object Editor)

**JOE** ist der **JobScheduler** Object Editor. Mit ihm können die **JobScheduler** Objekte (Jobs, Job-Ketten, Aufträge, Schedules, Prozess-Klassen, Sperren) konfiguriert werden.

Gestartet wird **JOE** mit dem Skript:

- `$SCHEDULER_HOME \bin\jobeditor.cmd` (Windows™)
- `$SCHEDULER_HOME /bin/jobeditor.sh` (Unix™)

## Schedule

Die Zeitsteuerung eines Jobs oder Auftrags kann direkt am Job oder Auftrag angegeben oder in einem Schedule ausgelagert werden. Dem Job oder Auftrag wird dann lediglich der Schedule zugewiesen. Falls mehrere Jobs oder Aufträge die gleiche Zeitsteuerung haben sollen, braucht man diese so nur einmal zu formulieren. Überdies können Schedules für einen bestimmten Zeitraum andere Schedules ersetzen. So erreicht man in der Verwendung von Schedules noch mehr Flexibilität bei der Zeitsteuerung.