



JOBSCHEDULER

JobScheduler - Installation Guide

Installation and Configuration

May 2013

Contact Information

Software- und Organisations-Service GmbH

Giesebrechtstr. 15
D-10629 Berlin
Germany

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

Last Updated: 05/24/2013 12:03 PM

This documentation is based on JobScheduler Version 1.3.12.3137.

Copyright © 2005-2013 SOS GmbH Berlin.

All rights reserved. All trademarks or registered trademarks are the property of their respective holders. All information and materials in this book are provided "as is" and without warranty of any kind. All information in this document is subject to change without further notice.

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.

Table of Contents

1 Installation	5
1.1 Requirements	5
1.2 Installation Using the Setup Program	6
1.3 Licenses	8
1.4 Installation Paths	9
1.5 Setup Packages	11
1.6 Setup Forms	12
1.6.1 The Basic JobScheduler Forms	13
1.6.2 The Update Service Package Form	16
1.6.3 The Database Support Package Forms	17
1.6.4 The Cron Job Package Form	21
2 Batch Installation	22
3 Database Configuration	27
3.1 MySQL	27
3.2 MS SQL Server and Sybase	27
3.3 PostgreSQL	27
3.4 Manual Creation of Database Table	28
4 Directory Structure after Installation	29
5 Starting and Stopping the JobScheduler	33
5.1 JobScheduler Demon on Unix	33
5.2 JobScheduler Service for Windows	33
6 Open JOC, JOE and JID	35
6.1 Open JOC (JobScheduler Operations Center)	35
6.2 Open JOE (JobScheduler Object Editor)	35
6.3 Open JID (JobScheduler Information Dashboard)	36
7 Configuration	37
7.1 The sos.ini File	37
7.2 The factory.ini File	37
7.3 The scheduler.xml File	37
7.4 The custom.js File	37
7.5 The hibernate.cfg.xml File	38
7.6 The jobscheduler_environment_variables.(sh cmd) File	38
7.7 The jobeditor_environment_variables.(sh cmd) File	38
7.8 The dashboard_environment_variables.(sh cmd) File	38
8 Update the JobScheduler	40
9 Multiple Installation	41
10 Installation of a Cluster	42
11 Deinstallation	43
11.1 Removal Using the Uninstaller	43
11.2 Manual Removal on Windows	43
11.3 Manual Removal on Unix	44
12 Automatic Update Procedure	45

13 Operation on 64bit Systems46
13.1 JOE and JID on 64bit Systems46
14 Troubleshooting47
15 Glossary48

1 Installation

The following steps should be carried out when making a new installation of the **JobScheduler**, in the order presented below:

Database Configuration (page 27) (optional, recommended)

The **JobScheduler** can be used without a database but it is strongly recommended that the **JobScheduler** use a database. The **JobScheduler** needs the database to archive logs and to remember **Job** and **Order** states.

The component **JID (JobScheduler Information Dashboard)** requires the database.

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. database systems are supported by the **JobScheduler**.

Because of licensing restrictions when used with MySQL, Sybase or MS SQL server databases, a JDBC driver appropriate to the database version used must be provided by the end users themselves. Alternatively, a jTDS JDBC driver, delivered with the **JobScheduler** setup, can be used for MS SQL Server and Sybase databases. Drivers for Oracle, PostgreSQL, Firebird and DB2 are delivered with the **JobScheduler** setup.

JobScheduler Installation (page 6)

Installation of the **JobScheduler** is carried out using a setup program. This can be downloaded from <http://www.sos-berlin.com> for Windows and Linux. For Solaris, HP-UX (Itanium) and AIX you get the **JobScheduler** on demand.

Unterstützte Betriebssysteme sind Windows 2003/XP(SP2)/Vista/2008/7, Linux starting with kernel 2.4, Solaris sparc 8/9/10 and AIX 6/7.

1.1 Requirements

- Oracle™ Java Runtime Environment (JRE) 32-Bit at least version 1.6.x. For AIX you can use the IBM™ Java 32-Bit at least version 1.6.x, too.
- For Unix:

A shell in /bin/sh (or a symlink)

The **JobScheduler** requires some 32-Bit libraries. These are on Linux:

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

Two components (**JOE (JobScheduler Object Editor)** and **JID (JobScheduler Information Dashboard)**) of the **JobScheduler** are SWT applications which requires an X Windows system and GTK2. The installation includes a 32-Bit swt.jar, so you need the 32-Bit libraries:

- `libgtk-x11-2.0.so.0`

- `libXtst.so.6`
- For Windows:

If you use Java 1.6 then the library `msvcr71.dll` must be stored in `C:\Windows\system32` and also in `C:\Windows\SysWOW64` on 64-Bit Windows.

If you use Java 1.7 then the library `msvcr100.dll` must be stored in `C:\Windows\system32` and also in `C:\Windows\SysWOW64` on 64-Bit Windows.

You find the `msvcr71.dll` or `msvcr100.dll` in the Java installation `[jre install path]\bin\plugin2`.
- If you use "Remote Configuration" then the Workload **JobSchedulers** and its Supervisor **JobScheduler** should have the same version. Further information about "Remote Configuration" can be found [here](#).

1.2 Installation Using the Setup Program

The following archive files including installer are available:

- `scheduler_linux.[release].tar.gz` for Linux
- `scheduler_win32.[release].zip` for Windows
- `scheduler_solarisx86.[release].tar.gz` for Solaris
- `scheduler_solaris-sparc.[release].tar.gz` for Solaris
- `scheduler_hpux-ia64-32.[release].tar.gz` for HP-UX Itanium
- `sscheduler_aix32.[release].tar.gz` for AIX

Unpack the archive in an arbitrary directory and change to the extracted directory `./jobscheduler.[release]`.

The installer can be started as a dialog or in batch mode (see [Batch Installation](#) (page 22)). If you use the installer as a dialog on Unix then it requires an X-Server. If an X-Server is not installed, then use the [Batch Installation](#).

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

Example: Start installer on Unix

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

Example: Start installer on Windows

The setup requires administrator privileges on Windows. The setup opens a dialog for this on Windows if necessary. On Unix a sudo prompt will be open. Don't log in as root on Unix but use sudo!

Under Unix, the root privileges are not required. If you want to install the **JobScheduler** without root privileges, then call

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

Example: Start installer on Unix without root permissions

The setup dialog starts with the selection of the language to be used in the setup. This is followed by a greeting, acceptance of the license conditions and the specification of two installation directories. The binaries and libraries are stored under the first path. The configuration and log files are stored under the second path.

For the rest of this documentation the first installation directory will be referred to as `$SCHEDULER_HOME` and the second as `$SCHEDULER_DATA`. Specification of the installation directories is followed by the [Package Selection](#) (page 11) dialog.

The forms which are subsequently presented for the configuration of the **JobScheduler** depend on the packages which are selected for installation alongside the **JobScheduler**. Further details of the **JobScheduler** configuration are to be found in the [Setup Forms](#) (page 12) chapter. After selection of the required packages, the necessary files are copied into the installation directories. After this, the scripts that configure the installation packages are executed. The processing of the installation scripts run during the setup is logged. This log file is to be found in the folder `$SCHEDULER_DATA/logs` and is named `Install_V[release]_[date][time]_[series number].log`.

JOC (JobScheduler Operations Center) can be accessed after the setup by entering the following URL in a web browser (Internet Explorer and Firefox are supported):

`http://localhost:[port]`

where [port] is the port specified for the **JobScheduler** during setup.

1.3 Licenses

The **JobScheduler** is available with a dual licensing model. The GNU GPL 2.0 license is available for Windows and Linux, otherwise the commercial license is required.



If you choose the commercial license then an input field is shown to enter the license key. The license key will be written in the file `$SCHEDULER_DATA/config/sos.ini`. Even so the license key is invalid you can continue the installation and edit the `$SCHEDULER_DATA/config/sos.ini` later.

1.4 Installation Paths

The setup knows two paths. Both paths are expanded with the **JobScheduler** ID as subdirectory. The form to enter the **JobScheduler** ID will be described later on.



The binaries and libraries are stored in this first path (`$SCHEDULER_HOME`). The default is

- `/opt/sos-berlin.com/jobscheduler` for Unix. If you use the installer without root permissions then you must choose another folder (e.g. `/home/[user]/sos-berlin.com/jobscheduler`).
- `C:\Program Files\sos-berlin.com\jobscheduler` for Windows



The configuration and log files are stored in this second path (`$SCHEDULER_DATA`). The default is

- `/home/[user]/sos-berlin.com/jobscheduler` for Unix
- `C:\ProgramData\sos-berlin.com\jobscheduler` for Windows Vista/2008/7
- `C:\Documents and Settings\All Users\Application Data\sos-berlin.com\jobscheduler` for Windows XP/2003

1.5 Setup Packages

The following packages may be selected during setup:

JobScheduler

This is the basic package and must be installed. The package contains **JOC (JobScheduler Operations Center)** which is a Ajax based interface for monitoring and controlling the **JobScheduler** objects, like *jobs*, *job chains* and *orders*. Further the package contains **JOE (JobScheduler Object Editor)** to create, edit and maintain the **JobScheduler** objects and **JID (JobScheduler Information Dashboard)**, which provides an overview of the *jobs* planned and those that have successfully been completed.

Update Service

This package inserts a *job* which checks from time to time (e.g. once a week) whether a new release is available.

Database Support

This package allows the job protocols to be stored in a database. 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. are supported.

Housekeeping Jobs

Housekeeping *Jobs* are automatically carried out by the **JobScheduler**, for example, resending temporarily stored protocol mails after a mail server failure; deleting temporary files or restarting the **JobScheduler** automatically. In addition, the Housekeeping *Jobs* package enables the **JobScheduler** to be configured as an event handler.

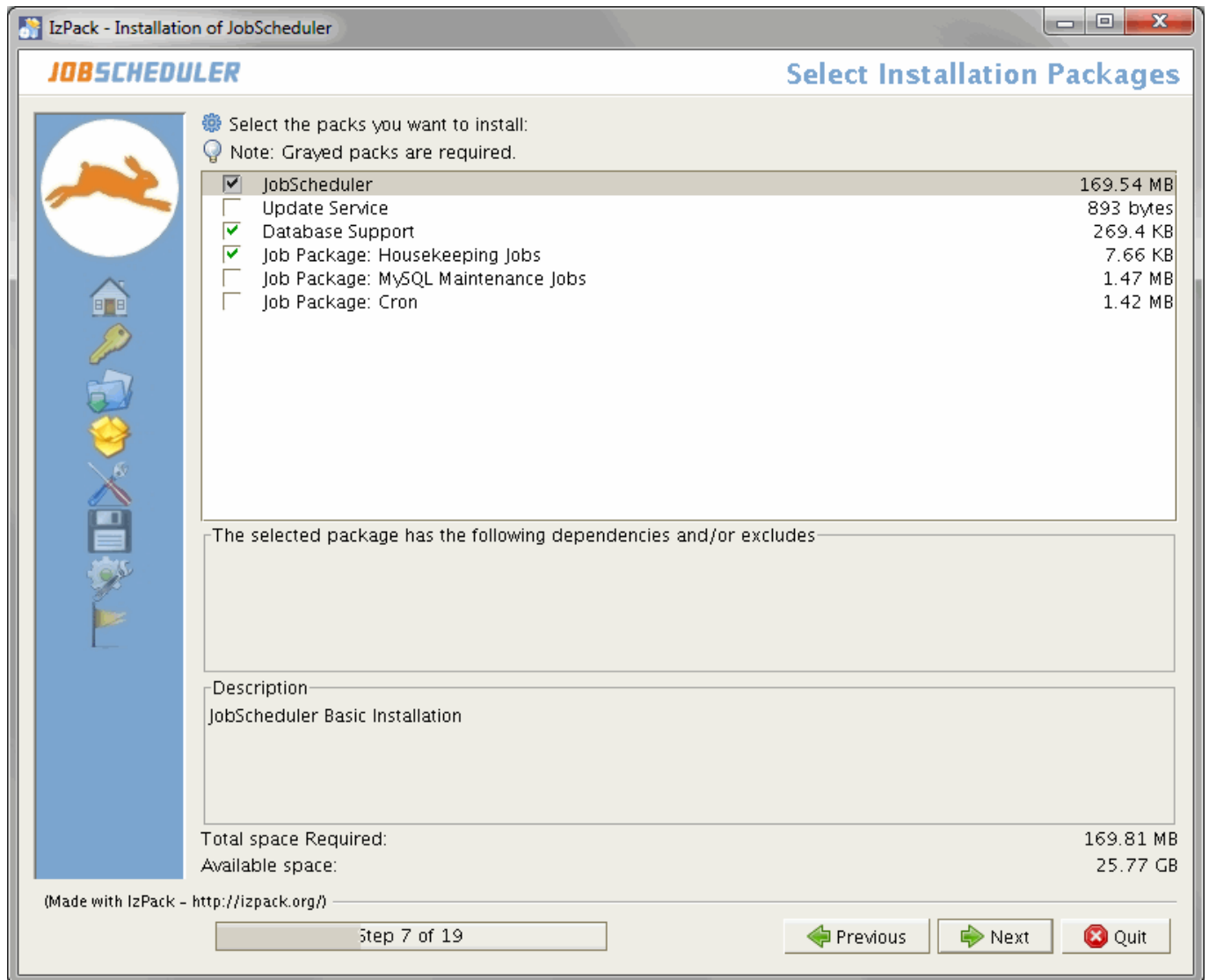
MySQL Maintenance Jobs

The MySQL *Jobs* package contains *jobs* for monitoring database replication. A MySQL database is required for the use of this package.

Cron Job

The Cron Adapter *Job* can be used to configure the **JobScheduler** with a crontab file. For this purpose, the *job* reads the crontab file and dynamically configures the **JobScheduler**. This package is only available for Unix systems.

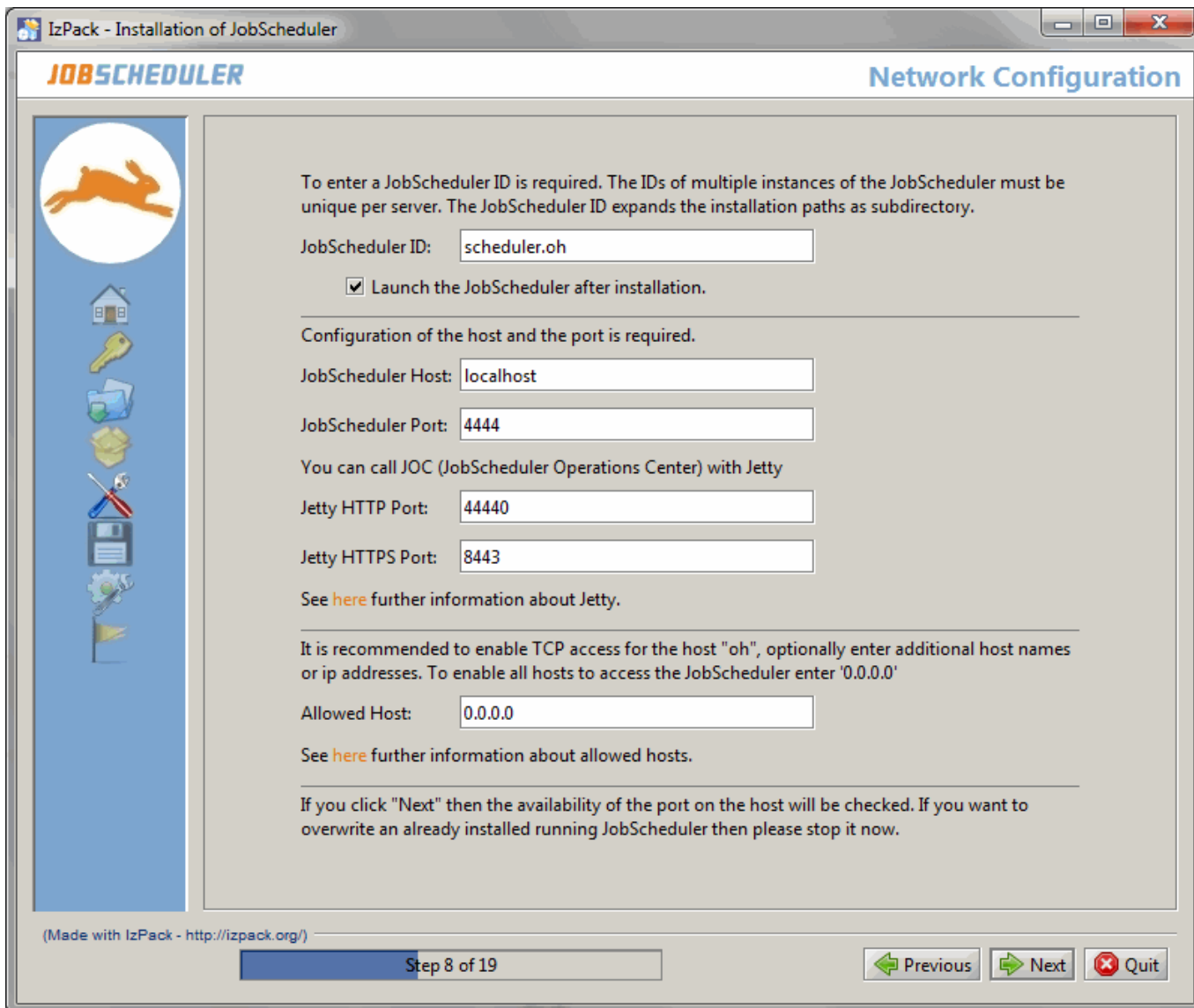
Package selection is made using the following dialog form:



1.6 Setup Forms

The number of forms shown during setup depends on the packages which have been chosen for installation.

1.6.1 The Basic JobScheduler Forms



JOBSCHEDULER 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

The **JobScheduler** ID is entered in the *JobScheduler ID* input box. Omit special characters like / \ : ; * ? ! \$ % & " < > () | ^

The ID is used on Windows for the name of the service after setup. The service name has the syntax `sos_scheduler_[scheduler_id]`.

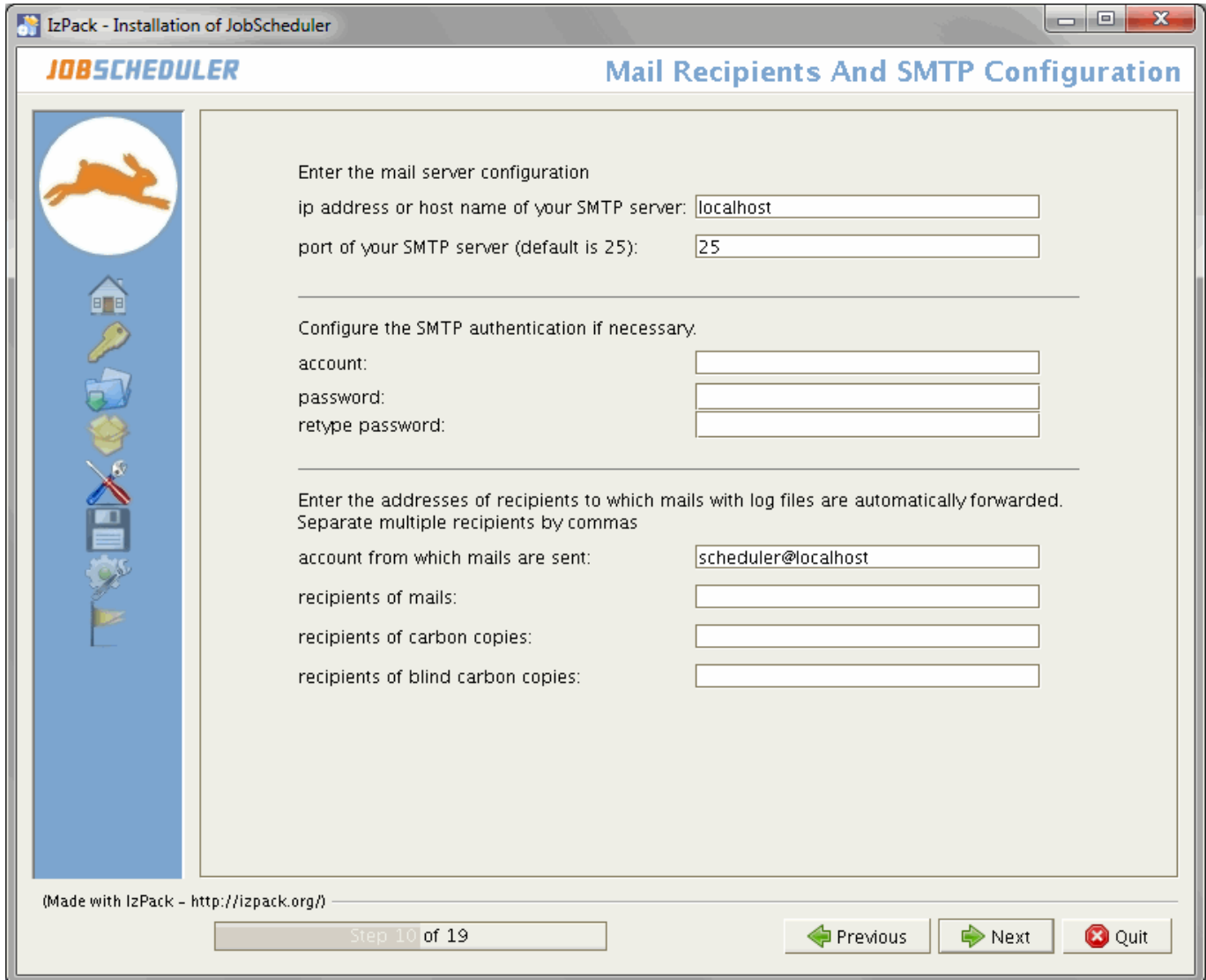
The *JobScheduler ID* must be unique over all installed **JobScheduler** (except you want to build a cluster). Further, the *JobScheduler ID* expands both installation paths as a subdirectory.

The next entry - the *JobScheduler Port* - is used for TCP communication with the **JobScheduler** e.g. for **JOC**.

You can call **JOC** with Jetty. Jetty needs a unique port for HTTP and HTTPS. See also [here](#) for more details about Jetty.

The *Allowed Host* field is required as a security feature of the **JobScheduler**, whereby communication can be restricted to particular computers. This is explained in more detail in the **JobScheduler** [documentation](#).

The *Port* and the *Allowed Host* entries are also written to the `$$SCHEDULER_DATA/config/scheduler.xml` file. The **JobScheduler** ID is written to the `$$SCHEDULER_HOME/bin/jobscheduler_environment_variables.(cmd|sh)` file. The ports for Jetty are written to the `$$SCHEDULER_DATA/config/jetty.xml` file. The configuration files can be [changed manually](#) (page 37) later on.



The SMTP Server is specified here along the mail sender, recipient and if required CC und BCC. Multiple addresses are to be separated by commas.

The values entered here configure the `$$SCHEDULER_DATA/config/factory.ini` file, which can also be [changed manually](#) (page 37) at a later date.



Configuration For Mail Notification/Event Handler

Choose the cases in which log files are forwarded by mail

- ☐ don't send mail with log file in case of error
- ☒ send mail with log file in case of error (recommended)
- ☒ don't send mail with log file in case of warnings
- ☐ send mail with log file in case of warnings (recommended)
- ☒ don't send mail with log file in case of success (recommended)
- ☐ send mail with log file in case of success

See [here](#) further information about the email notification settings.

☐ 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

See [here](#) further information about events in the JobScheduler.

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

Step 11 of 19

Previous Next Quit

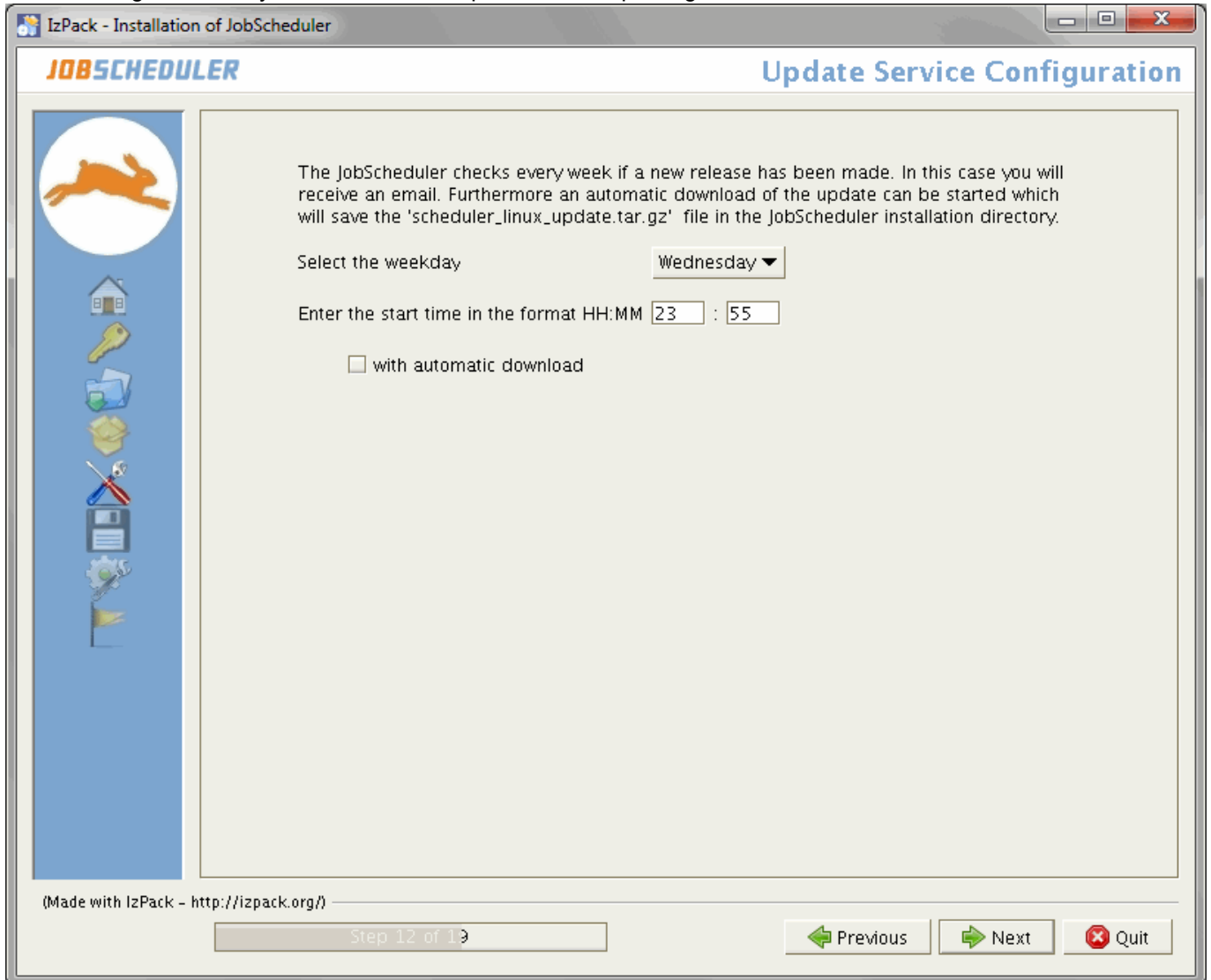
Specify information regarding whether the **JobScheduler** should automatically forward job log files by e-mail.

The entries made using this form are saved in the `$SCHEDULER_DATA/config/factory.ini` file, which can also be subsequently [changed manually](#) (page 37).

The lower part is only shown when the housekeeping package was selected. It enables the **JobScheduler** to be configured as an event handler. Corresponding objects will be created in `$SCHEDULER_DATA/config/live/sos/events` respectively. This is explained in more detail in the [Events documentation](#).

1.6.2 The Update Service Package Form

The following form is only shown when the Update Service package was selected.



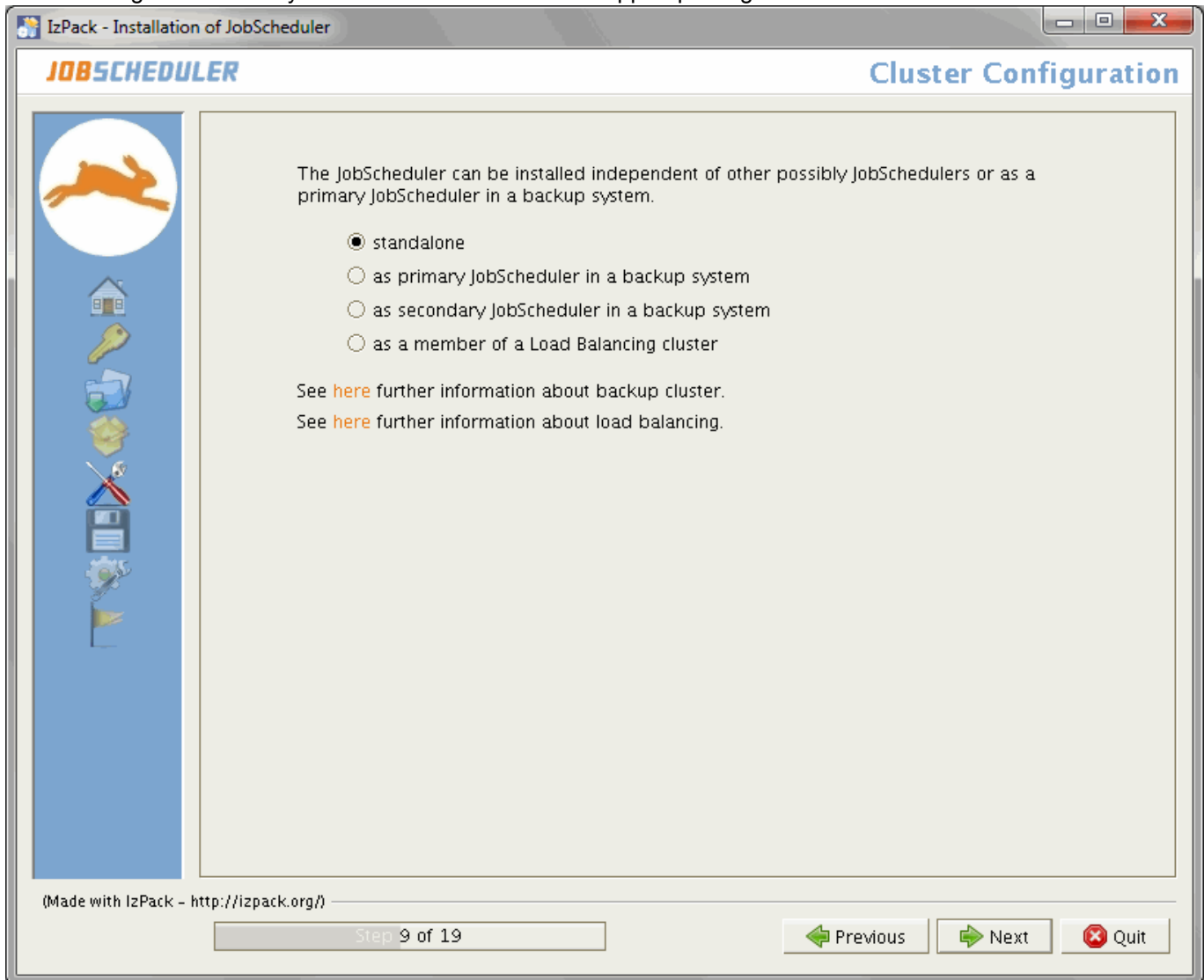
A **JobScheduler** *job* is added which checks every week if a new release has been made. You can assign three parameters to this *job*. The weekday, the time on which the *job* starts and if an automatic download can take place. In case of the automatic download, a file will be saved in `$SCHEDULER_DATA` with the operating system dependent name of

- `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`

The `$$CHEDULER_DATA/config/live/sos/update/scheduler_check_updates.job.xml` file may be used for later job configuration. Further information about the Update Service can be found in the [Update Service documentation](#).

1.6.3 The Database Support Package Forms

The following forms are only shown when the Database Support package was selected.



The screenshot shows the 'Cluster Configuration' window of the JobScheduler installation. The window title is 'IzPack - Installation of JobScheduler'. The main content area has a header 'JOB SCHEDULER' on the left and 'Cluster Configuration' on the right. Below the header, there is a vertical sidebar with icons representing different components. The main area contains the following text:

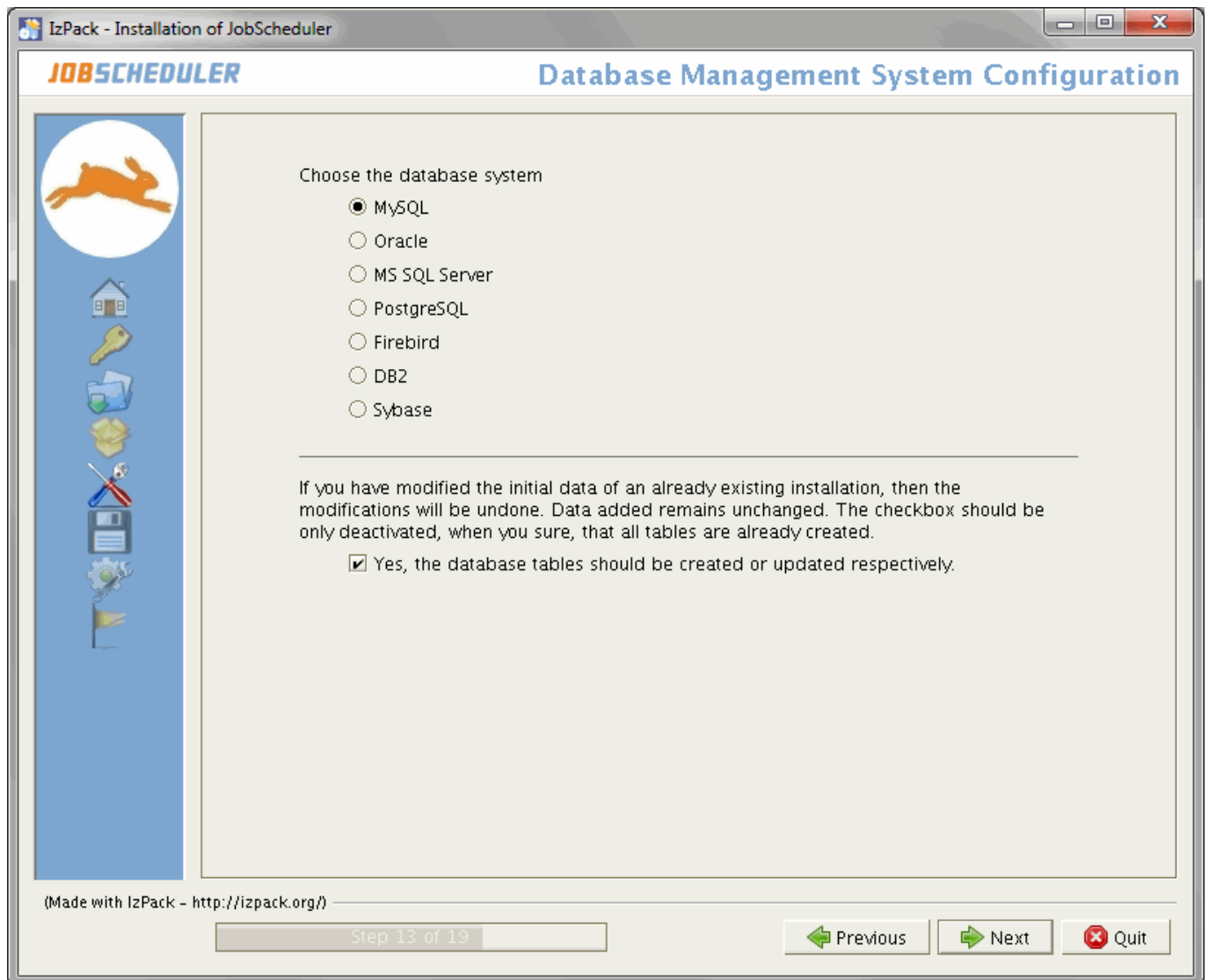
The JobScheduler can be installed independent of other possibly JobSchedulers or as a primary JobScheduler in a backup system.

- ☒ 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.

At the bottom, there is a status bar showing '(Made with IzPack - http://izpack.org/)', 'Step 9 of 19', and navigation buttons: 'Previous', 'Next', and 'Quit'.

The radio buttons in the form shown above determine whether the **JobScheduler** should be installed "stand-alone" or in a backup system or a load balancing cluster (see also [Installation of a Cluster](#) (page 42)). Further information about Backup Cluster can be found [here](#), about Load Balancing look [here](#).



IzPack - Installation of JobScheduler

JOBSCHEDULER Database Management System Configuration

Choose the database system

- ☒ MySQL
- ☐ Oracle
- ☐ MS SQL Server
- ☐ PostgreSQL
- ☐ Firebird
- ☐ DB2
- ☐ Sybase

If you have modified the initial data of an already existing installation, then the modifications will be undone. Data added remains unchanged. The checkbox should be only deactivated, when you sure, that all tables are already created.

☒ Yes, the database tables should be created or updated respectively.

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

Step 13 of 19

Previous Next Quit

The database system is specified in the upper selection on this form. It is recommended that the box in the center of the form is checked, so that a script which creates and fills the necessary database tables can be executed. Alternatively, the tables can be [created manually](#) (page 28). If you have already installed another **JobScheduler** with the same database connection then abandon this option.



JOB SCHEDULER Database Connection Configuration

Enter the database access parameters.

Host:

Port:

Database:

User:

Password:

Retype password:

For licence reasons MS SQL Server JDBC Drivers are not provided. Alternatively you can use the JTDS JDBC Driver which is provided.

☒ Yes, the JTDS JDBC Driver should be used.

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

Step 14 of 19

Previous Next Quit

The database connection information is specified in the input fields. The middle part where you can choose the provided jTDS JDBC driver is only shown for Sybase and MS SQL Server. If the jTDS JDBC driver is unchecked then you must enter your own JDBC driver in a next dialog.

This configuration is saved in the `$SCHEDULER_DATA/config/factory.ini`, `$SCHEDULER_DATA/config/hibernate.cfg.xml` and `$SCHEDULER_DATA/config/sos_settings.ini` files. All files can be [changed manually](#) (page 37) if required.

here for example.' Further down is a paragraph: 'If you click "Next" then a database connection test will be executed. If it fails then the setup omits all database transactions and the JobScheduler will not be started, but the database connection data will be configured. You can setup the database later and after this call the script ./install/scheduler_install_tables.sh to create and initialise the tables.' At the bottom, it says '(Made with IzPack - http://izpack.org/)', a progress bar showing 'Step 15 of 19', and buttons for 'Previous', 'Next', and 'Quit'." data-bbox="68 114 930 610"/>

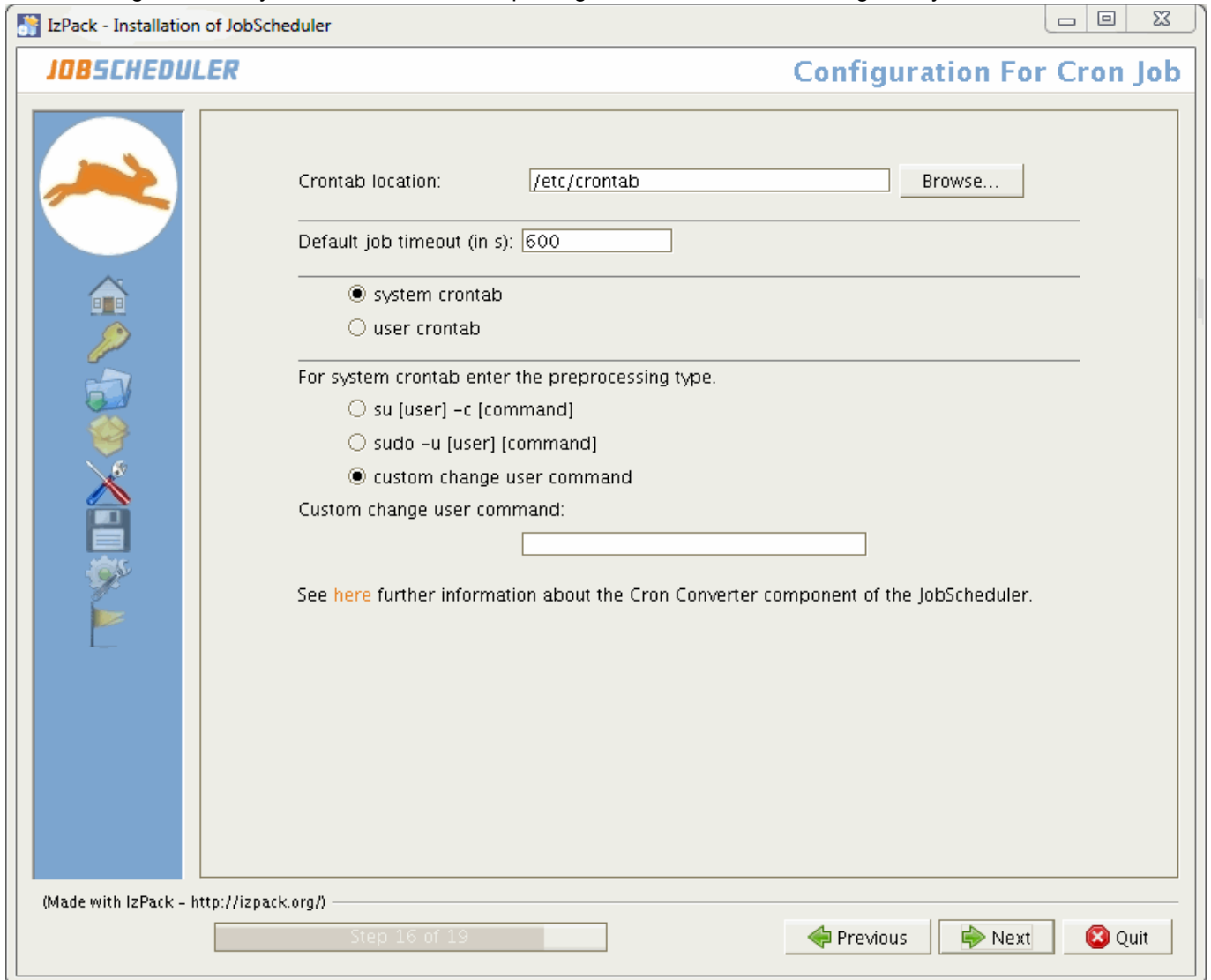
This dialog form is only offer for MySQL or you have unchecked the JTDS JDBC driver for Sybase and MS SQL Server databases. The script for the creation of the database tables is started by the setup program and requires a JDBC driver appropriate to the database system being used. The drivers for Oracle, PostgreSQL, Firebird and DB2 are included in the setup. However, because of licensing restrictions, the relevant MySQL, Sybase and MS SQL Server JDBC driver must be manually specified here. Note that for MS SQL Server and Sybase databases the JTDS JDBC driver that is delivered as part of the **JobScheduler** setup can be used when the appropriate checkbox in the previous form is activated.

As this driver will also be required by the **JobScheduler** later on, it is copied by the setup into the `$SCHEDULER_HOME/lib` folder.

If the Firebird database system is being used, then it is important that no other connections to the database server exist during installation.

1.6.4 The Cron Job Package Form

The following form is only shown when the Cron package was selected. This dialog is only available for Unix.



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

Further information about the Cron Converter can be found [here](#).

2 Batch Installation

Note that when the **JobScheduler** installation is started from a parameterized XML file, no dialog forms will appear. The file `scheduler_install.xml` included in the extracted directory `./jobscheduler.[release]` can be used for this purpose. You can start the setup as follows after you have edited this file with the installation paths, host, port, database connection, etc..

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

Example: Start installer on Unix with parameter file

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

Example: Start installer on Windows with parameter file

The setup requires administrator privileges on Windows. The setup opens a dialog for this on Windows if necessary. On Unix a sudo prompt will be open. Don't log in as root on Unix but use sudo!

Under Unix, the root privileges are not required. If you want to install the **JobScheduler** without root privileges, then call

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

Example: Start installer on Unix with parameter file and without root privileges

The `scheduler_install.xml` has the following content:

```
<?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!
         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:]<
/UserPathPanelElement>

</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: MySQL Maintenance Jobs
    The job package for MySQL includes jobs for monitoring of replications.
    MySQL database support is required to operate this feature. -->
    <pack index="6" name="MySQL" selected="false"/>

    <!-- 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 to 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 and MS SQL Server JDBC Drivers are
    not provided. Alternatively you can use 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"/>

  </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
    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>

```

Example: scheduler_install.xml

This XML file mirrors all the values which can specified during a setup dialog.

3 Database Configuration

It is recommended that the **JobScheduler** is allocated a database and/or database schema and a database user. Instructions for the creation of the database itself are to be taken from the database documentation. 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 and Sybase ASE 15.0. are supported. The **JobScheduler** setup program creates the necessary database tables if the [Database Support](#) (page 11) package is installed and the database connection is specified in the appropriate setup form.

The setup also requires a sufficient database permissions for CREATE TABLE, INSERT, UPDATE, DELETE und SELECT statements.

The database configuration information is saved in the following [configuration files](#) (page 37).

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

3.1 MySQL

Because of licensing restrictions a MySQL JDBC Driver is not provided. Please download a MySQL JDBC driver before you start the **JobScheduler** installer.

3.2 MS SQL Server and Sybase

Because of licensing restrictions when used with Sybase or MS SQL server databases, a JDBC driver appropriate to the database version used must be provided by the end users themselves. Alternatively, a jTDS JDBC driver, delivered with the **JobScheduler** setup, can be used for MS SQL Server and Sybase databases. Otherwise please download a MS SQL or Sybase JDBC driver before you start the **JobScheduler** installer.

3.3 PostgreSQL

PostgreSQL requires PL/SQL. Check the languages that are available for your database by using

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

where "postgres" is the user name and "scheduler" is the database name. "createlang" is available from the PostgreSQL `bin` directory. Should "plpgsql" not be listed in the output of this command then please enable this language by

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

The following two PostgreSQL server variables must have the following values

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

For all those that do not want to change this globally, this setting can be changed on a per user level:

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

Example: Set `standard_conforming_strings` and `bytea_output` per user level

where "scheduler" is the user name of the **JobScheduler** database.

3.4 Manual Creation of Database Table

SQL scripts which create the database tables required by the **JobScheduler** are available, should they not have been correctly created by the setup program. These scripts can be run using `$SCHEDULER_HOME/install/scheduler_install_tables.(sh|cmd)`.

Ensure that the database connection is correctly entered in the `$SCHEDULER_DATA/config/sos_settings.ini` [configuration file](#) (page 37).

4 Directory Structure after Installation

The contents of some of the following directories depend on the packages installed during setup and on the operating system used. In such cases the package name and/or operating system is noted in brackets after the directory or file name. Should a package name or an operating system be specified for a directory, then all the files in the directory will share this dependency.

The following directory structure should be found in `$SCHEDULER_HOME`:

- + `bin` (Windows)
 - `dashboard.cmd` Start script for **JID**
 - `hostjava.dll` Program library
 - `hostole.dll` Program library
 - `jobeditor.cmd` Start script for **JOE**
 - `jobscheduler.cmd` Start script for the **JobScheduler**
 - `jobscheduler_environment_variables.cmd` Script to set the **JobScheduler** environment
 - `jobscheduler_event.cmd` Event handling script
 - `jobscheduler_client.pl` Perl script (TCP/UDP client for sending XML commands to a **JobScheduler**)
 - `scheduler.exe` **JobScheduler** engine
 - `scheduler.exe.local` File for local usage of DLLs
 - `spidermonkey.dll` JavaScript (Mozilla) program library
- + `bin` (Unix)
 - `dashboard.sh` Start script for **JID**
 - `jobeditor.sh` Start script for **JOE**
 - `jobscheduler.sh` Start script for the **JobScheduler**
 - `jobscheduler_environment_variables.sh` Script to set the **JobScheduler** environment
 - `jobscheduler_event.sh` Event handling script
 - `jobscheduler_client.pl` Perl script (TCP/UDP client for sending XML commands to a **JobScheduler**)
 - `scheduler` **JobScheduler** engine
 - `scheduler_safe.sh` Watchdog script to respawn the **JobScheduler**
 - `setuid` Program to process scripts in a different user context, see FAQ
- + `db` SQL files to create database tables
 - + `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` (see mssql directory)
 - + `procedures` (MySQL Maintenance Jobs)
 - `scheduler_job_procedure.sql`
 - `scheduler_user_jobs.sql` (MySQL Maintenance Jobs)
- + `oracle` Oracle 8.1.7, 9.2, 10g
 - `*.sql` (see mssql directory)
- + `fbsql` Firebird 1.5
 - `*.sql` (see mssql directory)
- + `pgsql` PostgreSQL 8.x
 - `*.sql` (see mssql directory)
 - `hibernate_sequence.sql`
- + `db2` IBM DB2 8
 - `*.sql` (see mssql directory)
- + `sybase` Sybase 15.x
 - `*.sql` (see mssql directory)
- + `doc` Documentations
- + `install`
 - `scheduler_install_tables.sh` Script for execute manually above SQL files (Unix)
 - `scheduler_install_tables.cmd` Script for execute manually above SQL files (Windows)
- + `lib`
 - `*.jar` Java archives (for Java jobs)
 - `scheduler.dll` for Java debugging (Windows)
 - `*.so`, `*.sl`, `*.a` libraries (Unix)
- + `operations_gui` HTML and Javascript files used by **JOC**
- + `Uninstaller` Program to uninstall the **JobScheduler**

- + **user_bin**
 - **dashboard_environment_variables.cmd-example** Script example to set the **JID** environment (Windows)
 - **jobeditor_environment_variables.cmd-example** Script example to set the **JOE** environment (Windows)
 - **jobscheduler_environment_variables.cmd-example** Script example to set the **JobScheduler** environment (Windows)
 - **dashboard_environment_variables.sh-example** Script example to set the **JID** environment (Unix)
 - **jobeditor_environment_variables.sh-example** Script example to set the **JOE** environment (Unix)
 - **jobscheduler_environment_variables.sh-example** Script example to set the **JobScheduler** environment (Unix)

The following directory structure should be found in **\$SCHEDULER_DATA**:

- + **config**
 - + **cache** Configuration directory in a Workload **JobScheduler** (Replicate of a Supervisor **JobScheduler** remote directory)
 - + **events** (Housekeeping jobs/event handling)
 - + **live** Local configuration directory for the **JobScheduler** (*Hot Folder*)
 - + **sos**
 - + **events** (Event Handler Jobs)
 - + **housekeeping** (Housekeeping Jobs)
 - + **mysql** (MySQL Maintenance Jobs)
 - + **update** (Update Service Job)
 - + **operations_gui**
 - **custom.js** Configuration file for **JOC**
 - + **remote** Local configuration directory for a Workload **JobScheduler** on a Supervisor **JobScheduler**
 - **factory.ini** **JobScheduler** configuration file
 - **hibernate.cfg.xml** Database connection for hibernate classes
 - **jetty.xml** Configuration file for Jetty
 - **scheduler.xml** **JobScheduler** configuration file
 - **scheduler.xsd** The XML schema definition for the **JobScheduler** configuration
 - **scheduler_mail.xsl** A stylesheet for emails with log files
 - **sos.ini** License file
 - **sos_settings.ini** Database connection
 - **web.xml** Configuration file for Jetty

- + [jobs](#) Documentations of JITL (**JobScheduler** Integrated Template Library)
- + [logs](#) Depository for log files

5 Starting and Stopping the JobScheduler

5.1 JobScheduler Demon on Unix

On Unix systems, the **JobScheduler** is operated as a demon. To start and stop the **JobScheduler** use the script:

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

Example: Starting the JobScheduler on Unix

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

Example: Stopping JobScheduler on Unix

In addition to start and stop, this script accepts additional parameters, e.g. debug, restart, abort and kill.

If you want the **JobScheduler** to be started automatically at server startup, then please copy this script and the `$SCHEDULER_HOME/bin/jobscheduler_environment_variables.sh` file to the appropriate startup/shutdown directory - usually this is `/etc/init.d`.

If the **JobScheduler** doesn't start then look into the `$SCHEDULER_DATA/logs/scheduler.log` for the reason. You may be missing dependent libraries. This can be checked with

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

5.2 JobScheduler Service for Windows

On Windows systems, the **JobScheduler** is installed as service. You can find the **JobScheduler** service by opening the Windows service panel and looking for a service with a name starting with "SOS **JobScheduler**". The service has the system account after the setup. The service is configured to be started automatically at server startup.

You can start and stop the service in the Windows service panel or with

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

Example: Starting the JobScheduler service on Windows

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

Example: Stopping the JobScheduler service on Windows

...where `[JobSchedulerId]` is the Id of the **JobScheduler**.

Further you can start the **JobScheduler** from the command line. Ensure that the service has not already been started and use the following script:

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

Example: Starting the JobScheduler on Windows

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

Example: Stopping the JobScheduler on Windows

In addition to start and stop, this script accepts additional parameters, e.g. debug, restart, abort and kill.

If the **JobScheduler** doesn't start then look into the [\\$SCHEDULER_DATA/logs/scheduler.log](#) for the reason.

6 Open JOC, JOE and JID

6.1 Open JOC (JobScheduler Operations Center)

JOC (JobScheduler Operations Center) is a GUI for monitoring and operating the **JobScheduler**. You open **JOC** in a browser (Internet Explorer and Firefox are supported) with the following URL.

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

You can open **JOC** also with Jetty with the following URLs.

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

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

When you open **JOC** from another computer, you must make sure that the communication is not blocked by a firewall or by the [Security setting](#) of the **JobScheduler**.

If you have updated the **JobScheduler** installation, it may be necessary to clear the browser cache for the changes to take effect in **JOC**.

6.2 Open JOE (JobScheduler Object Editor)

JOE (JobScheduler Object Editor) is an application to create, edit and maintain the **JobScheduler** objects (*jobs*, *job chains*, *orders* and *schedules*).

You start **JOE** with ...

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

Example: Starting JOE on Windows

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

Example: Starting the JOE on Unix

An X-Server and GTK2 is necessary on Unix systems. The necessary libraries must be installed in 32-bit.

When you start **JOE** on Windows, it may be that not happened. Then an initial fatal error occurred. Use debug as the argument of the call to see the error.

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

Example: Debug JOE on Windows

For example when you get the error

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

after starting **JOE** then you must adjust the environment variable `$JAVA_HOME` in `$SCHEDULER_HOME/user_bin/jobeditor_environment_variables.(sh|cmd)` (page 38).

6.3 Open JID (JobScheduler Information Dashboard)

JID (JobScheduler Information Dashboard) is an application to provide an overview of the *jobs* planned and those that have successfully been completed. See also the [Dashboard documentation](#) for more details.

You start **JID** with ...

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

Example: Starting JID on Windows

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

Example: Starting JID on Unix

An X-Server and GTK2 is necessary on Unix systems. The necessary libraries must be installed in 32-bit.

When you start **JID** on Windows, it may be that not happened. Then an initial fatal error occurred. Use debug as the argument of the call to see the error.

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

Example: Debug JOE on Windows

For example when you get the error

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

after starting **JID** then you must adjust the environment variable `$JAVA_HOME` in `$SCHEDULER_HOME/user_bin/dashboard_environment_variables.(sh|cmd)` (page 38).

7 Configuration

The **JobScheduler** is configured using the following files:

- `$SCHEDULER_DATA/config/sos.ini`
- `$SCHEDULER_DATA/config/factory.ini`
- `$SCHEDULER_DATA/config/scheduler.xml`
- `$SCHEDULER_DATA/config/operations_gui/custom.js` (for **JOC**)
- `$SCHEDULER_DATA/config/hibernate.cfg.xml` (database connection of **JID**)
- `$SCHEDULER_HOME/bin/jobscheduler_environment_variables.(sh|cmd)`

These files are configured during the **JobScheduler** setup, using the information entered at the time. In addition, you can create further three files to adjust the environment. These are:

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

7.1 The sos.ini File

In the file `$SCHEDULER_DATA/config/sos.ini` the license key is included. In addition, Java options and if necessary the location of the JVM can be set. Further details about the entries in this file are to be found [here](#).

7.2 The factory.ini File

E-mail settings, information about the database connection and the Java archives classpath are saved in the `$SCHEDULER_DATA/config/factory.ini` file. Further details about the entries in this file are to be found [here](#).

7.3 The scheduler.xml File

The **JobScheduler** port information are to be found in the `$SCHEDULER_DATA/config/scheduler.xml` file. Further details about this file are to be found [here](#).

7.4 The custom.js File

The `$SCHEDULER_DATA/config/operations_gui/custom.js` file is used by **JOC (JobScheduler Operations Center)**. Beside other settings you can configure the language and filters in particular. See also [here](#) for more details.

7.5 The hibernate.cfg.xml File

The `$$SCHEDULER_DATA/config/hibernate.cfg.xml` file is used by **JID (JobScheduler Information Dashboard)** to get the database connection. See also the [Dashboard documentation](#) for more details.

7.6 The jobscheduler_environment_variables.(sh|cmd) File

In this file the start parameter and the ID of the **JobScheduler** are set.

This file should not be changed because the changes after a **JobScheduler** Update may have been lost. If environment variables need to be adjusted, then please put them in a file `$$SCHEDULER_HOME/user_bin/jobscheduler_environment_variables.(sh|cmd)`. You can use the file `$$SCHEDULER_HOME/user_bin/jobscheduler_environment_variables.(sh|cmd)-example` as a template. `$$SCHEDULER_HOME/user_bin/jobscheduler_environment_variables.(sh|cmd)` must be executable on Unix.

The `$$SCHEDULER_HOME/bin/jobscheduler_environment_variables.sh` file is particularly relevant for Linux/Unix systems because the `LD_LIBRARY_PATH` or `$JAVA_HOME` is set, which must be customized, if the **JobScheduler** should not find the Java virtual machine.

In this case the following error is logged in `$$SCHEDULER_DATA/logs/scheduler.log`:

```
[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]]
```

If you modify the **JobScheduler** Id on Windows then note that the corresponding service must be reinstalled.

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

Example: Reinstall the JobScheduler service

7.7 The jobeditor_environment_variables.(sh|cmd) File

In this file the environment of **JOE (JobScheduler Object Editor)** can be set.

You start **JOE** with `$$SCHEDULER_HOME/bin/jobeditor.(sh|cmd)`. This file should not be changed because the changes after a **JobScheduler** Update may have been lost. If environment variables (e.g. `$JAVA_HOME`) need to be adjusted, then please put them in a file `$$SCHEDULER_HOME/user_bin/jobeditor_environment_variables.(sh|cmd)`. You can use the file `$$SCHEDULER_HOME/user_bin/jobeditor_environment_variables.(sh|cmd)-example` as a template. `$$SCHEDULER_HOME/user_bin/jobeditor_environment_variables.(sh|cmd)` must be executable on Unix.

7.8 The dashboard_environment_variables.(sh|cmd) File

In this file the environment of **JID (JobScheduler Information Dashboard)** can be set.

You start **JID** with `$$SCHEDULER_HOME/bin/dashboard.(sh|cmd)` but this file should not be changed because the changes after a **JobScheduler** Update may have been lost. If environment variables (e.g. `$JAVA_HOME`) need to be adjusted, then please put them in a file `$$SCHEDULER_HOME/user_bin/dashboard_environment_variables.(sh|cmd)`. You can use the file `$$SCHEDULER_HOME`

`/user_bin/dashboard_environment_variables.(sh|cmd)-example` as a template. `$$SCHEDULER_HOME`
`/user_bin/dashboard_environment_variables.(sh|cmd)` must be executable on Unix.

8 Update the JobScheduler

A special "Update-Setup" of the **JobScheduler** is available.

- `scheduler_linux_update.[release].tar.gz` for Linux
- `scheduler_win32_update.[release].zip` for Windows
- `scheduler_solarisx86_update.[release].tar.gz` for Solaris
- `scheduler_solaris-sparc_update.[release].tar.gz` for Solaris
- `scheduler_hpux-ia64-32_update.[release].tar.gz` for HP-UX Itanium
- `sscheduler_aix32_update.[release].tar.gz` for AIX

Call of this setup runs analogous to the "[Full-Setup](#)" (page 6).

You can also use the "Full-Setup" to update the **JobScheduler**, but is only practical with reservation, because if you want to update a **JobScheduler** installation where the first installation was done before release 1.3.9 then the directory tree is not the same. Instead of that please use the "Update-Setup".

9 Multiple Installation

You can install any number of **JobScheduler**.

The following points must be observed when completing the [Network Configuration](#) (page 13) form of the **JobScheduler** basic package setup:

- The **JobScheduler ID** should be unique amongst all the **JobScheduler** except you want to install a [cluster](#) (page 42).

On Windows the **JobScheduler** ID is used after the setup is completed to set the name of the **JobScheduler** service in the `sos_scheduler_[scheduler_id]` form.

- The *TCP port* must also be unique amongst all the **JobSchedulers** installed on one computer.

It is recommended that all **JobSchedulers** installed on a computer or in a network use the same database connection. This must be the case a Backup Cluster or Load Balancing is to be used.

10 Installation of a Cluster

Each **JobScheduler** in a cluster has almost the same setup configuration. Particularly, they have the same **JobScheduler** Id and the same database connection. After the installation of the first **JobScheduler** of the cluster the file `$SCHEDULER_HOME/scheduler_install.xml` will be created. An easy way to install the other clustered **JobScheduler** is to use this file for a [Batch-Installation](#) (page 22). You must only edit the `schedulerHost` value in the `scheduler_install.xml` and in addition the `clusterOptions` when building a Backup Cluster. The value of `databaseCreate` should be set to off, as the database has already been created when the primary **JobScheduler** was set up.

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

Example: Snippet of `$SCHEDULER_HOME/scheduler_install.xml`

Further information about Backup Cluster can be found [here](#), about Load Balancing look [here](#).

11 Deinstallation

11.1 Removal Using the Uninstaller

The Uninstaller `$SCHEDULER_HOME/Uninstaller/uninstall.jar` is initialized by the setup program used to install the **JobScheduler**. The Uninstaller is started using:

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

Example: Start uninstaller on Unix

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

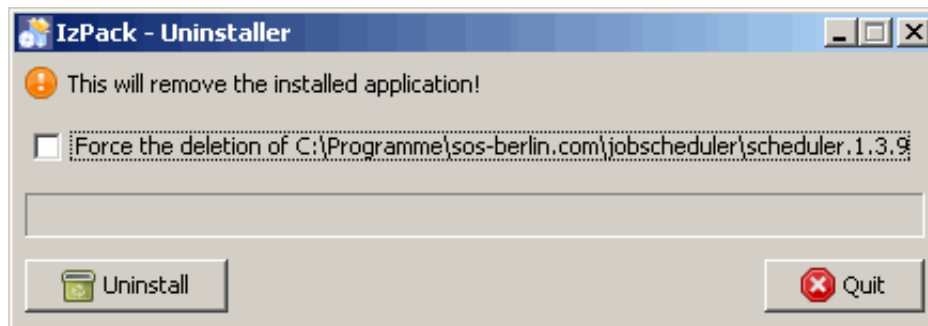
Example: Start uninstaller on Windows

On Unix the uninstaller ask for root permissions via sudo. If you have installed the **JobScheduler** without root permission then the uninstaller doesn't need root permission. In this case you type:

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

Example: Start uninstaller on Unix without root permissions

On Windows a dialog box asks for the removal of the **JobScheduler** to be confirmed.



A database created for the **JobScheduler** must be deleted manually.

The "SOS **JobScheduler** id=[scheduler_id]" service on Windows should be removed manually after uninstalling a **JobScheduler**. It is important to note here the correct [scheduler_id] - that is the ID specified during installation of the **JobScheduler**. It may be that this service is marked as being deactivated. In this case, the service will only be removed after the computer has been restarted. This can be verified by opening the service panel (Start->Run services.msc) or by entering:

Should the service only have been deactivated, then a renewed installation of a **JobScheduler** with the same [scheduler_id] will only be possible after the computer has been restarted.

11.2 Manual Removal on Windows

To manually remove a **JobScheduler**, it is necessary to open a shell (Start->Run cmd) and then carry out the following steps.

- Stop the **JobScheduler**

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

An error message will be shown, should the **JobScheduler** already have been stopped. This message can be ignored.

- Remove the **JobScheduler** Service

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

- Remove the database

The documentation for any database which may have been installed for the **JobScheduler** should be consulted for instructions as to its removal.

- Deregister the hostole.dll program library

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

- Delete all files and directories

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

11.3 Manual Removal on Unix

To manually remove the **JobScheduler**, a shell should be opened and then the following steps carried out.

- Stop the **JobScheduler**

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

An error message will be shown, should the **JobScheduler** already have been stopped. This message can be ignored.

- Remove the database

The documentation for any database which may have been installed for the **JobScheduler** should be consulted for instructions as to its removal.

- Delete all files and directories

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

12 Automatic Update Procedure

A web service has been installed on <http://www.sos-berlin.com>, which answers queries about the most recent available version (release) of the **JobScheduler**. Should such a release be available, then this information will be conveyed to the query initiator.

A *job* is delivered with the **JobScheduler** when the packages Update Service was chosen which asks this web service once a week if a newer version of the **JobScheduler** has been released. Should a newer release be available, then an e-mail will be sent to the system administrator, informing him about this. If required, the *job* can also automatically download the necessary files.

More details are to be found in [Update Service documentation](#).

13 Operation on 64bit Systems

The **JobScheduler** is implemented as a 32-Bit application. You can operate the **JobScheduler** in a 64-Bit environment on the supported platforms. The **JobScheduler** requires furthermore a 32-Bit JRE and some 32-Bit libraries (see [requirements](#) (page 5)).

If you have a 64-Bit JRE installed then it is possible that the **JobScheduler** setup use it and writes the path of the 64-Bit JRE installation into the `$JAVA_HOME` variable in the `$SCHEDULER_HOME/bin/jobscheduler_environment_variables.(sh|cmd)`. In this case the **JobScheduler** doesn't start and you must change the `$JAVA_HOME` in `$SCHEDULER_HOME/user_bin/jobscheduler_environment_variables.(sh|cmd)` (page 38).

If the **JobScheduler** doesn't start then look into the `$SCHEDULER_DATA/logs/scheduler.log` for the reason. You may be missing dependent libraries. This can be checked with

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

13.1 JOE and JID on 64bit Systems

It can happen that neither **JOE** nor **JID** can start because a 64-Bit JRE is called. In this case you must change the `$JAVA_HOME` variable in `$SCHEDULER_HOME/user_bin/jobeditor_environment_variables.(sh|cmd)` (page 38) and `$SCHEDULER_HOME/user_bin/dashboard_environment_variables.(sh|cmd)` (page 38).

JOE and **JID** can be started with a 64-Bit JRE. The following steps are necessary.

- Download a 64-Bit [swt.jar](#) and copy it to `$SCHEDULER_HOME/lib`.
- Set `$JAVA_HOME=[/path/to/64-Bit JRE]` in `$SCHEDULER_HOME/user_bin/jobeditor_environment_variables.(sh|cmd)` and `$SCHEDULER_HOME/user_bin/dashboard_environment_variables.(sh|cmd)`.
- The following libraries must have 64-Bit on Unix.
 - `libgtk-x11-2.0.so.0`
 - `libXtst.so.6`

14 Troubleshooting

Assistance in troubleshooting issues can be found in the [Home Page](#), in our [FAQ](#) or on the [Sourceforge forum](#).

Glossary

Job Chains

A series of jobs that process orders one after the other. The **JobScheduler** starts the jobs in a job chain automatically, once a order has been started for the chain. Job chains allow a number of orders to be processed in parallel, by starting multiple instances of jobs (tasks).

Jobs

Programs and scripts that are executed by the **JobScheduler** have to be embedded in jobs. Jobs can contain either start executable files or contain job scripts that use the **JobScheduler** program interface. More than one instance of a job (task) may run at any one time, should this be required to scale performance.

There are two types of jobs: standalone and order jobs. Whereas order jobs are started by orders within a job chain, standalone jobs can be started independently: either manually, scheduled or by directory monitoring. Standalone jobs cannot be run in job chains.

JOC (JobScheduler Operations Center)

JOC (JobScheduler Operations Center) is the **JobScheduler** interface for monitoring and controlling **JobScheduler** objects such as jobs, job chains and orders.

JOC is opened in a web browser using the address `http://[scheduler_host]:[scheduler_port]`, where `[scheduler_host]` and `[scheduler_port]` are the host name and the TCP ports number of the **JobScheduler** (e.g. `http://localhost:4444`).

JOE (JobScheduler Object Editor)

JOE is the **JobScheduler** Object Editor. This is used to configure **JobScheduler** objects (jobs, job chains, orders, schedules, process classes and locks).

JOE is started using the script:

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

Orders

Orders activate the processing of job chains. Orders may also contain parameters for the jobs in a job chain. Every job in a job chain has access to the order parameters. Order parameters overwrite job parameters of the same name. Orders can be started according to time.

An order processes the jobs in a job chain one after the other. Orders can be configured so that, if a error in processing a job occurs, the order ...

- is removed from the job chain;
- continues with a further job in the chain;
- continues with the job that caused the initial error being repeated
- stands still - that is the order processing is suspended until it is restarted manually.

Schedules

Time-based starting of jobs or orders can either be directly specified for each job or order or can be delegated to a schedule. Individual jobs or orders are then referred to this schedule. This means that if several jobs or orders have the same start parameters, these need only be specified once in the schedule. In addition, one schedule can be replaced by another for a particular period of time, thereby increasing the flexibility of setting job and order start parameters.