Job Scheduling

JOB SCHEDULER

Installation and Configuration

January 2011

Contact Information

Software- und Organisations-Service GmbH
Giesebrechtstr. 15
10629 Berlin
Germany

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

Last Updated: January 2011

Table of Contents

1 Installation

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

- **Database Configuration** (page 22) (optional)

  The Job Scheduler can be used without a database, which, however, means that job protocols and job histories will be only stored to disk. A further advantage of a database is that it allows the *Job Scheduler Web Interface (page 8)* to be be used, which in turn allows old job protocols to be retrieved.

  Further, the choice of additional packages which can be installed alongside the Job Scheduler is restricted when database support is not selected.

  MySQL, Oracle, Microsoft SQL Server, PostgreSQL, Firebird, DB2 and Sybase database systems are supported by the Job Scheduler.

  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 Job Scheduler setup, can be used for MS SQL Server and Sybase databases. Drivers for Oracle, PostgreSQL, Firebird and DB2 are delivered with the Job Scheduler setup.

  Note that the notices Troubleshooting (page 44) chapter about the choice of an appropriate JDBC driver, the configuration of a MySQL database server in ANSI mode are important and should be read and the requirement of PL/SQL for PostgreSQL.

- **Job Scheduler Installation** (page 4)

  Installation of the Job Scheduler is carried out using a setup program which can be downloaded from http://www.jobscheduler.de. Windows 2000/2003/XP/Vista, Linux starting with kernel 2.4, Solaris 8/9/10, HP-UX (Itanium) 11.23/11.31 and AIX 5.2/5.3 operating systems are supported.
  The *Job Scheduler* since release 1.3.7 is not available under the GNU GPL 2.0 License for Solaris, HP-UX and AIX.
  Notification: Windows 2000 and XP(SP1) are not supported approximately from the middle of 2011.

- **Web Server Configuration** (page 37) (optional)

  The use and configuration of a web server for the *Job Scheduler* is only necessary if the *Web Interface (page 8)* package is selected during installation. This package contains an extended web interface for PHP.

1.1 Requirements

- Sun 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)

1.2 Installation Using the Setup Program

The following archive files are available for download from http://www.jobscheduler.de:

- **scheduler_linux.[release].tar.gz** for Linux (archive with setup program)

- **scheduler_solarisx86.[release].tar.gz** for Solaris (archive with setup program)
- **scheduler_solaris-sparc.[release].tar.gz** for Solaris (archive with setup program)
- **scheduler_win32.[release].zip** for Windows (archive with setup program)
- **scheduler_hpux-ia64-32.[release].tar.gz** for HP-UX Itanium (archive with setup program)
- **scheduler_aix32.[release].tar.gz** for AIX (archive with setup program)

One of the following setup programs will be found after unpacking the relevant archive:

- **scheduler_linux32.jar** for Linux
- **scheduler_solarisx86.jar** for Solaris
- **scheduler_solaris-sparc.jar** for Solaris
- **scheduler_win32.jar** for Windows
- **scheduler_hpux-ia64-32.jar** for HP-UX Itanium
- **scheduler_aix32.jar** for AIX.

The setup program can be started as a dialog or in batch mode (see batch installation (page 24)).

The "*jar*" programs are started using:

```
windows-shell>java -jar [download_path]\scheduler_win32.jar
```
or
```
windows-shell>[download_path]\setup.cmd
```

```
unix-shell>sudo java -jar [download_path]/scheduler_[unix_os](64-)32.jar
```
or
```
unix-shell>[download_path]/setup.sh
```

where [download_path] is the location of the "*jar*" program.

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

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 *[install_path]* and the second as *[appdata_path]*. Specification of the installation directories is followed by the Package Selection (page 8) dialog.

The forms which are subsequently presented for the configuration of the *Job Scheduler* depend on the packages which are selected for installation alongside the *Job Scheduler*. Further details of the *Job Scheduler* configuration are to be found in the Setup Forms (page 10) 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 *[appdata_path]/logs* and is named *Install_V[release]_[date][time]_[series number].log*.

The *Job Scheduler* OperationsGui can be accessed after setup by entering the following URL in a web browser (Internet Explorer starting with version 5.5 and Firefox are supported):

```
http://localhost:[port]
```

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

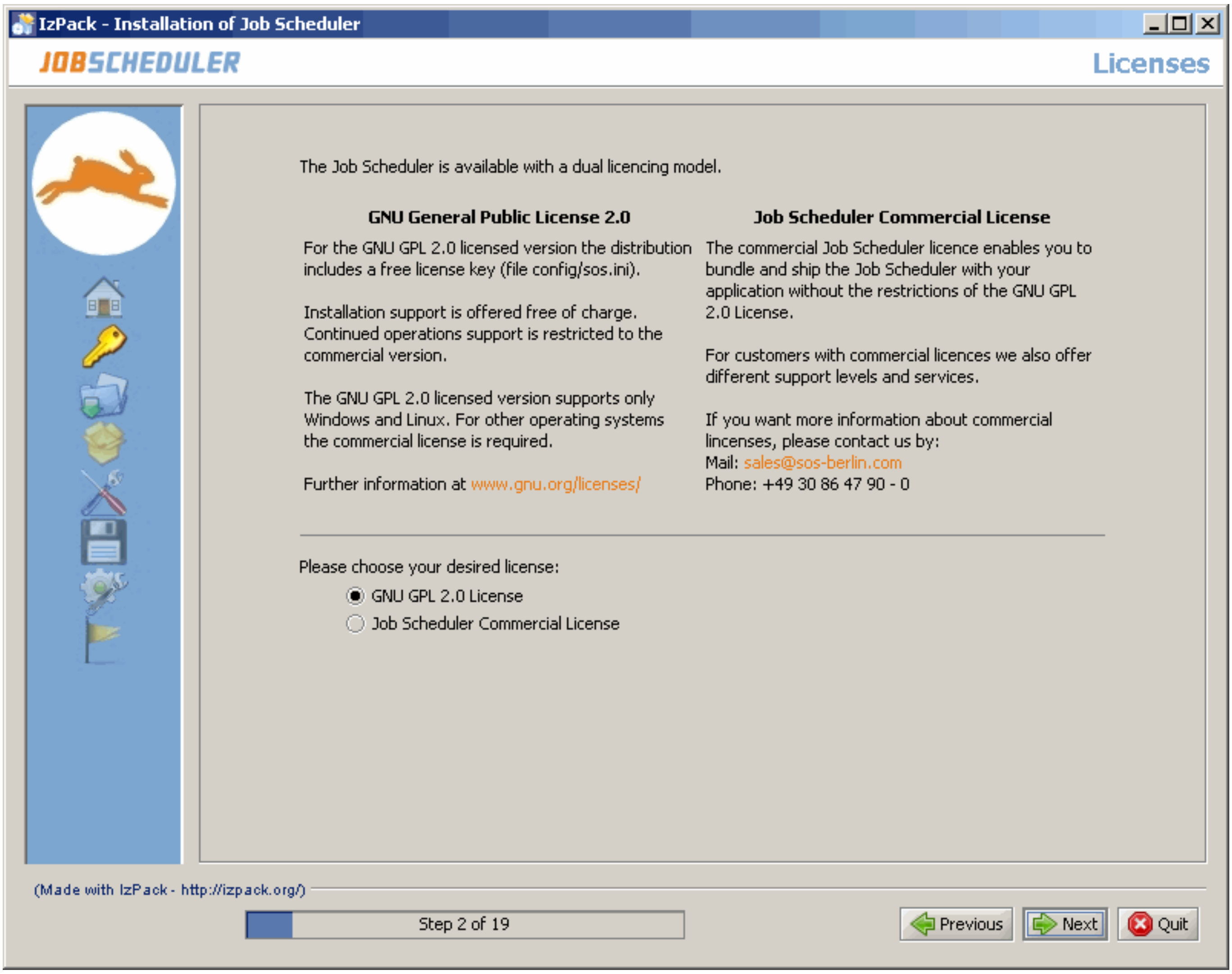
For Unix Users
The setup is a dialog program and requires that an X-Server is installed. If an X-Server is not installed, then use the Batch Installation (page 24).
The following libraries are required by the Job Scheduler:

- [install_path]/lib/libstdc++.so.6.0.8, [install_path]/lib/libgcc_s.so.1 (Linux)
- [install_path]/lib/libstdc++.so.6.0.8, [install_path]/lib/libgcc_s.so.1 (Solaris)
- [install_path]/lib/libstdc++.so.6.9, [install_path]/lib/libgcc_s.so.0 (HP-UX Itanium)
- [install_path]/lib/libstdc++.a, [install_path]/lib/libgcc_s.a (AIX)

These libraries are included in the setup. It is important to ensure that all the dependent libraries in the distribution are installed. This is, for example, the case with SUSE 9.

1.3 Licenses

The Job Scheduler is available with a dual licencing model. The GNU GPL 2.0 license is available for Windows and Linux, otherwise the commercial lisense is required.



The Job Scheduler is available with a dual licencing model.

GNU General Public License 2.0

For the GNU GPL 2.0 licensed version the distribution includes a free license key (file config/sos.ini).

Installation support is offered free of charge. Continued operations support is restricted to the commercial version.

The GNU GPL 2.0 licensed version supports only Windows and Linux. For other operating systems the commercial license is required.

Further information at www.gnu.org/licenses/

Job Scheduler Commercial License

The commercial Job Scheduler licence enables you to bundle and ship the Job Scheduler with your application without the restrictions of the GNU GPL 2.0 License.

For customers with commercial licences we also offer different support levels and services.

If you want more information about commercial lincenses, please contact us by:
Mail: sales@sos-berlin.com
Phone: +49 30 86 47 90 - 0

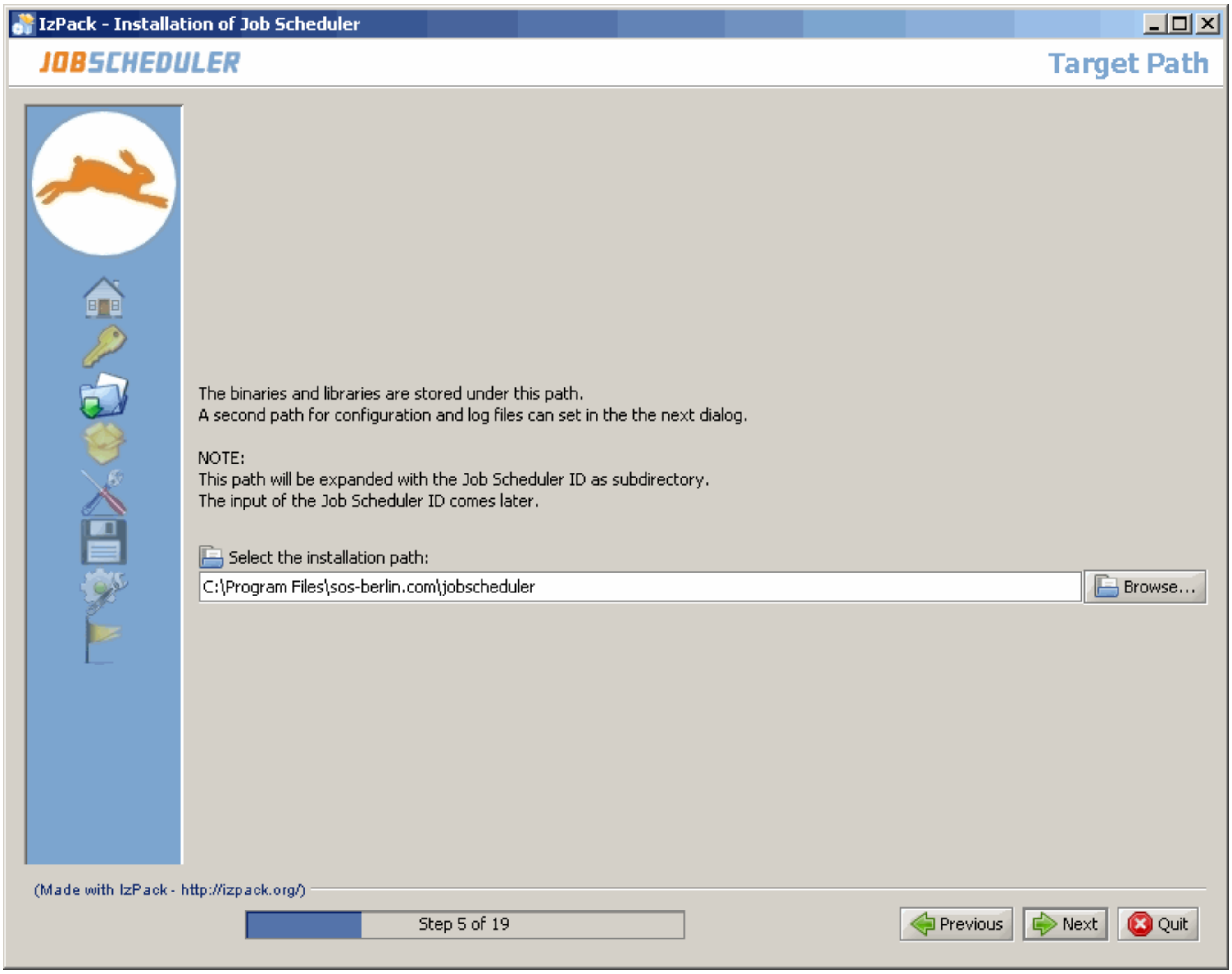
Please choose your desired license:

● GNU GPL 2.0 License
○ Job Scheduler Commercial License

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 `[appdata_path]/config/sos.ini`. Even so the license key is invalid you can continue the installation and edit the `[appdata_path]/config/sos.ini` later.

1.4 Installation Paths

The setup knows two paths. Both paths are expanded with the *Scheduler ID* as subdirectory. The form to enter the *Scheduler ID* comes later.


IzPack - Installation of Job Scheduler
JOBSCHEDULER
Target Path

The binaries and libraries are stored under this path.
A second path for configuration and log files can set in the the next dialog.

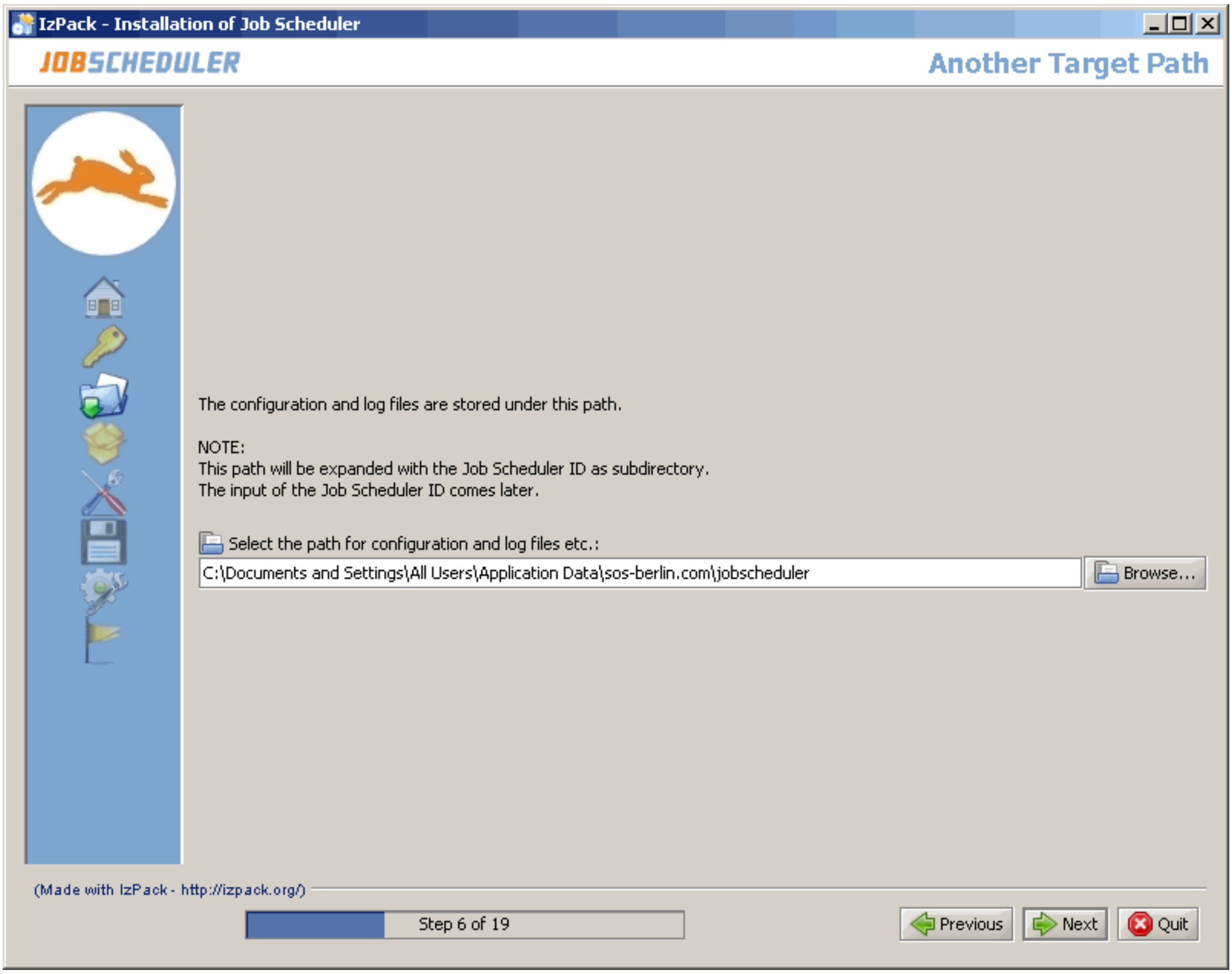
NOTE:
This path will be expanded with the Job Scheduler ID as subdirectory.
The input of the Job Scheduler ID comes later.

Select the installation path:
C:\Program Files\sos-berlin.com\jobscheduler

Browse...
(Made with IzPack - http://izpack.org/)
Step 5 of 19
Previous
Next
Quit

The binaries and libraries are stored in this first path. The default is

- `/opt/sos-berlin.com/jobscheduler` for Unix
- `C:\Program Files\sos-berlin.com\jobscheduler` for Windows



IzPack - Installation of Job Scheduler

JOBSCHEDULER Another Target Path

The configuration and log files are stored under this path.

NOTE:
This path will be expanded with the Job Scheduler ID as subdirectory.
The input of the Job Scheduler ID comes later.

Select the path for configuration and log files etc.:
C:\Documents and Settings\All Users\Application Data\sos-berlin.com\jobscheduler

The configuration and log files are stored in this second path. The default is

- `/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

1.5 Setup Packages

The following packages may be selected during setup:

- **Job Scheduler**
  This is the basic package and must be installed. This package contains the OperationsGui which is a web interface for monitorning and controlling the *Job Scheduler* objects like jobs and orders. Further the package contains the *Job Scheduler* Object Editor (JOE) to configure the *Job Scheduler* objects.

- **Update Service**
  This package inserts a job which checks every week if a new release has been made.

- **Database Support**

This package allows the job history and job protocols to be saved in a database. MySQL, Oracle, SQL Server, PostgreSQL, Firebird, Sybase and DB2 databases are supported.

- **Web Interface**
  In the Web Interface you can call a view of past jobs beside the OperationsGui. The package requires that PHP version 4.3 or higher is installed (for Firebird support use version 5.0 or higher).
  The language of the setup program also sets the language for the PHP web interface. Use of this package requires a database.

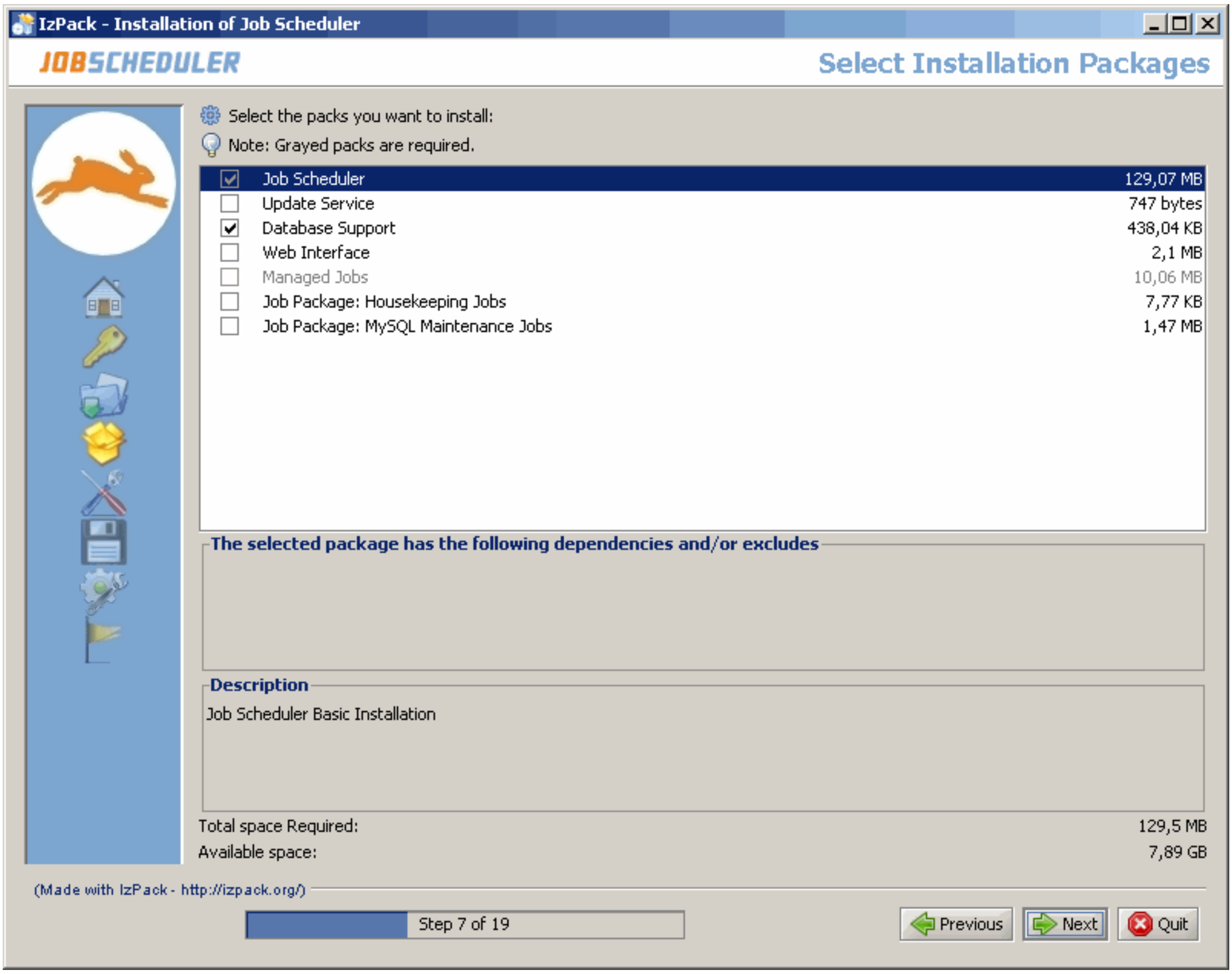
- **Housekeeping Jobs**
  Housekeeping jobs are automatically carried out by the Job Scheduler, for example, resending temporarily stored protocol mails after a mail server failure; deleting temporary files or restarting the *Job Scheduler* automatically.
  In addition, the Housekeeping Jobs package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

- **Managed Jobs**
  Managed Jobs is an extention of the PHP web interface within which you can configure *Job Scheduler* objects. It is possible to monitor and configure multiple *Job Scheduler*. For more information please look into the Managed Jobs documentation.
  Further database tables are created. Note that the language selected for the setup affects the language used in the SETTINGS table and therefore the content of the settings listed in the web interface. This can, however, be changed at a later point using the `[install_path]/bin/import_settings.sh [language]` script or `[install_path]/bin/import_settings.cmd [language]`. In both cases the `[language]` is set using either *de* for German or *en* for English (lower case). Use of this package requires a database.

- **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 *Job Scheduler* with a crontab file. For this purpose, the job reads the crontab file and dynamically configures the *Job Scheduler*. This package is only available for Unix systems.

Package selection is made using the following dialog form:



IzPack - Installation of Job Scheduler
JOBSCHEDULER
Select Installation Packages

Select the packs you want to install:

Note: Grayed packs are required.

☑	Job Scheduler	129,07 MB
□	Update Service	747 bytes
☑	Database Support	438,04 KB
□	Web Interface	2,1 MB
□	Managed Jobs	10,06 MB
□	Job Package: Housekeeping Jobs	7,77 KB
□	Job Package: MySQL Maintenance Jobs	1,47 MB

The selected package has the following dependencies and/or excludes

Description

Job Scheduler Basic Installation

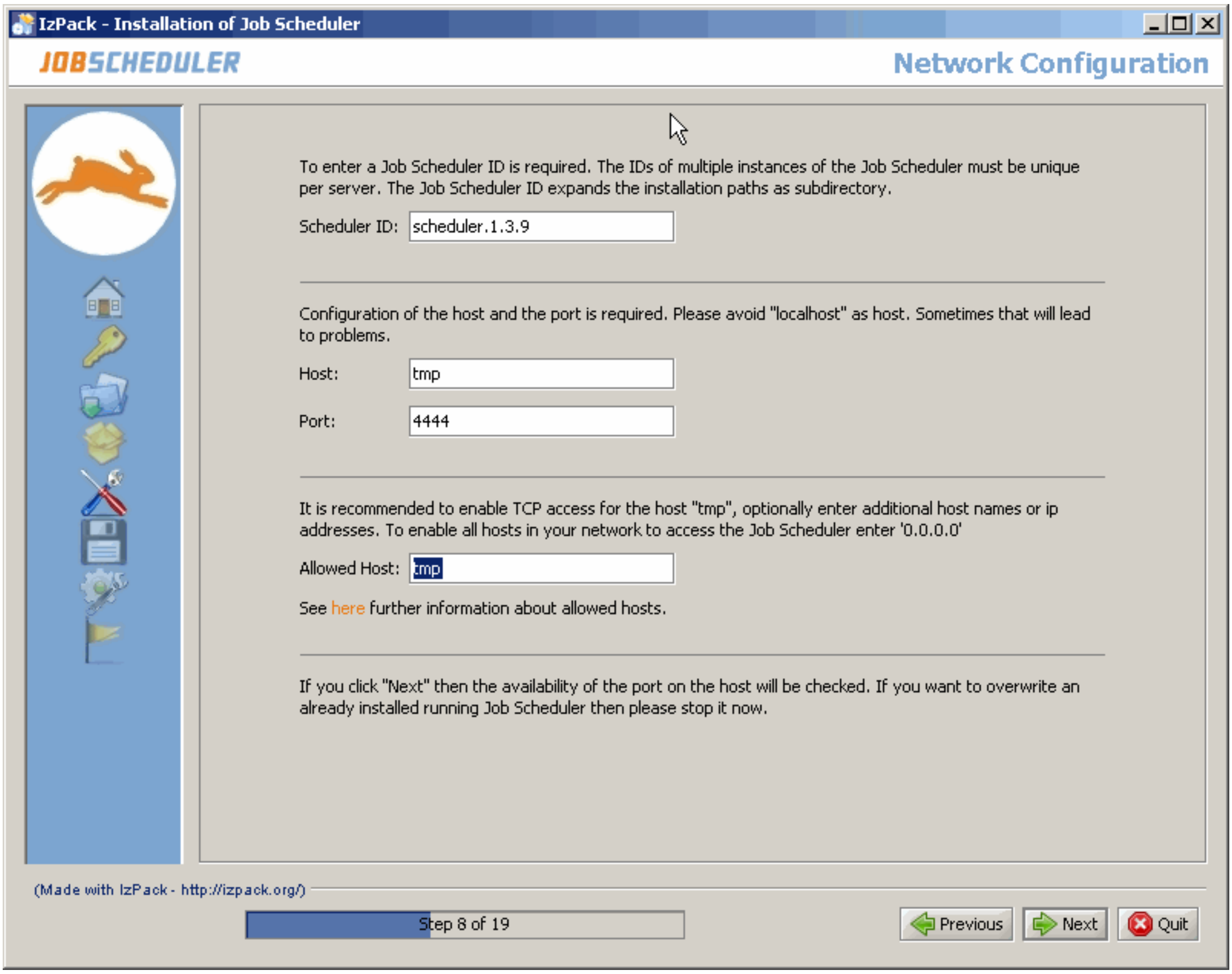
Total space Required: 129,5 MB

Available space: 7,89 GB

1.6 Setup Forms

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

1.6.1 The Basic Job Scheduler Forms


IzPack - Installation of Job Scheduler
JOBSCHEDULER
Network Configuration
(Made with IzPack - http://izpack.org/)
Step 8 of 19
Previous
Next
Quit

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

Scheduler ID: scheduler.1.3.9

Configuration of the host and the port is required. Please avoid "localhost" as host. Sometimes that will lead to problems.

Host: tmp

Port: 4444

It is recommended to enable TCP access for the host "tmp", optionally enter additional host names or ip addresses. To enable all hosts in your network to access the Job Scheduler enter '0.0.0.0'

Allowed Host: tmp

See here further information about allowed hosts.

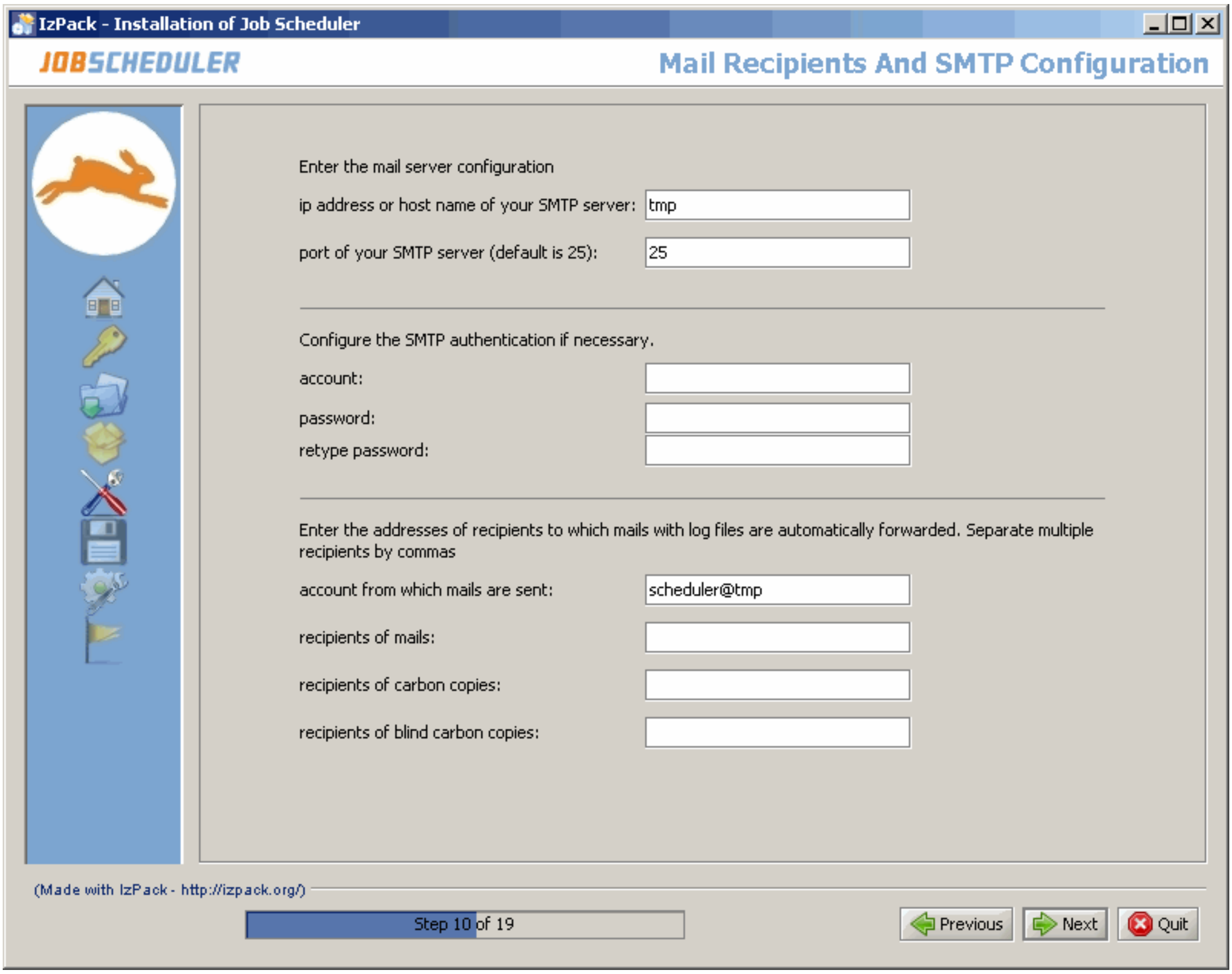
If you click "Next" then the availability of the port on the host will be checked. If you want to overwrite an already installed running Job Scheduler then please stop it now.

The Job Scheduler ID is entered in the *Scheduler 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 next entry - the *Port* - is used for communication with the *Job Scheduler* i.e. for the OperationsGui.

The *Allowed Host* field is required as a security feature of the *Job Scheduler*, whereby communication can be restricted to particular computers. This is explained in more detail in the *Job Scheduler* documentation.

The entries made for host and port configure the *[install_path]/web/custom/custom.inc.php* file. The Port and the Allowed Host entries are also written to the *[appdata_path]/config/scheduler.xml* file. The Scheduler ID is written to the *[install_path]/bin/jobscheduler_environment_variables.(cmd|sh)* file. The configuration files can be changed manually (page 35) later on.


IzPack - Installation of Job Scheduler
JOBSCHEDULER
(Made with IzPack - http://izpack.org/)
Step 10 of 19
Previous
Next
Quit

Mail Recipients And SMTP Configuration

Enter the mail server configuration

ip address or host name of your SMTP server: tmp

port of your SMTP server (default is 25): 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: scheduler@tmp

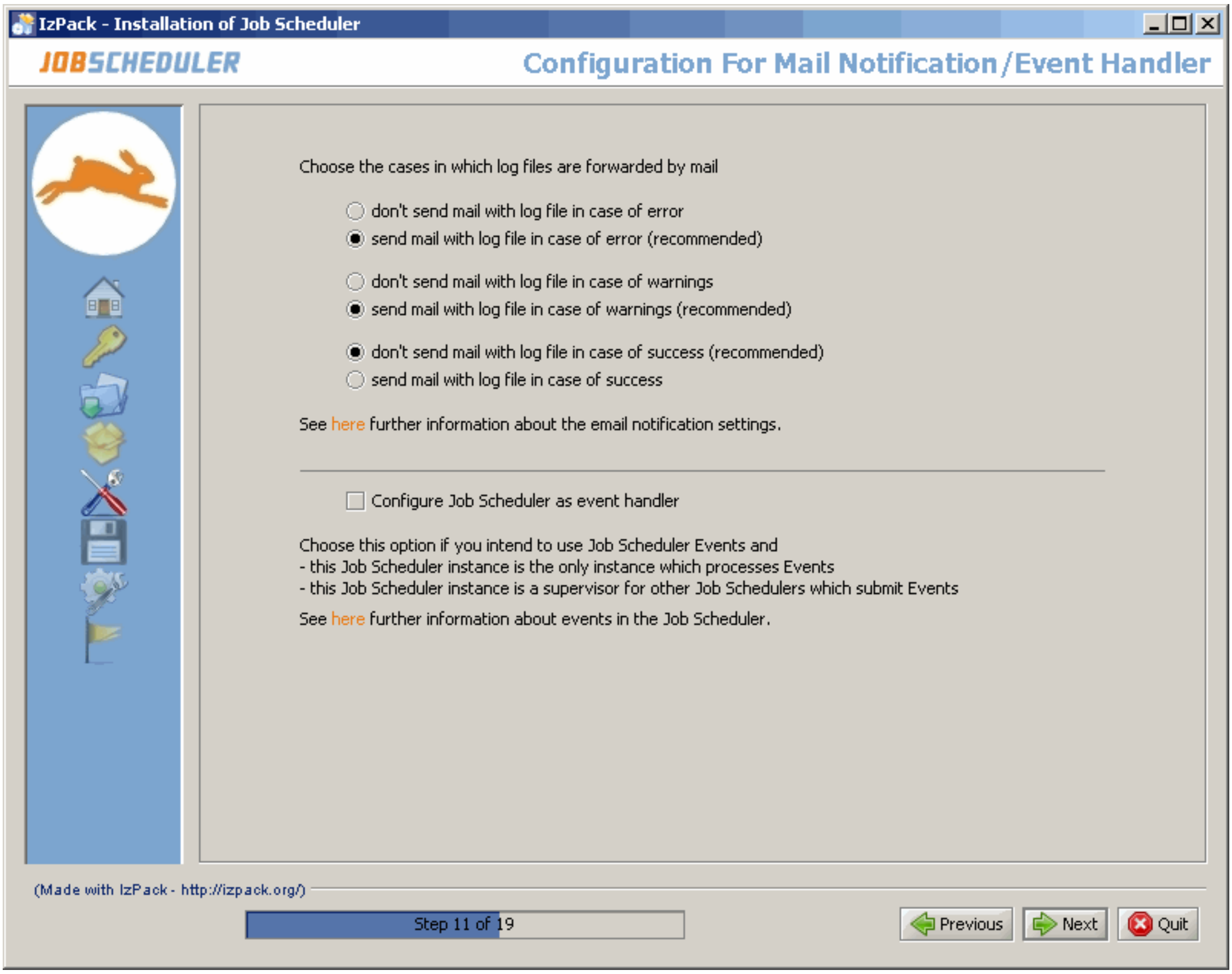
recipients of mails:

recipients of carbon copies:

recipients of blind carbon copies:

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 *[appdata_path]/config/factory.ini* file, which can also be changed manually (page 35) at a later date.



IzPack - Installation of Job Scheduler
JOBSCHEDULER
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 Job Scheduler as event handler

Choose this option if you intend to use Job Scheduler Events and
- this Job Scheduler instance is the only instance which processes Events
- this Job Scheduler instance is a supervisor for other Job Schedulers which submit Events

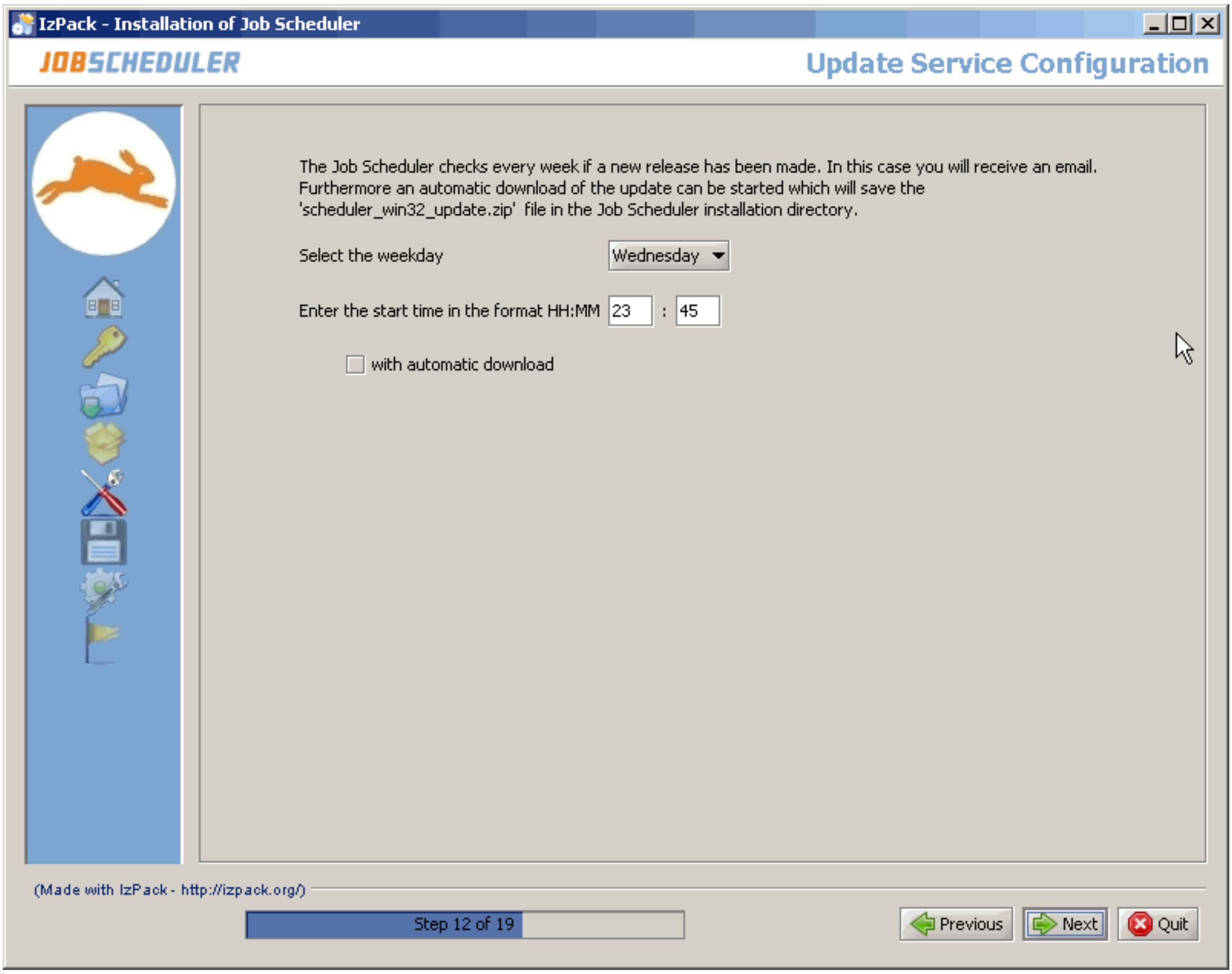
See here further information about events in the Job Scheduler.

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

The entries made using this form are saved in the *[appdata_path]/config/factory.ini* file, which can also be subsequently changed manually (page 35).

The lower part is only shown when the housekeeping package was selected. It enables the *Job Scheduler* to be configured as an event handler. Corresponding objects will be created in [appdata_path]/config/live/sos/events respectively.

1.6.2 The Update Service Package Form



IzPack - Installation of Job Scheduler
JOBSCHEDULER
Update Service Configuration

The Job Scheduler 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_update.zip' file in the Job Scheduler installation directory.

Select the weekday Wednesday

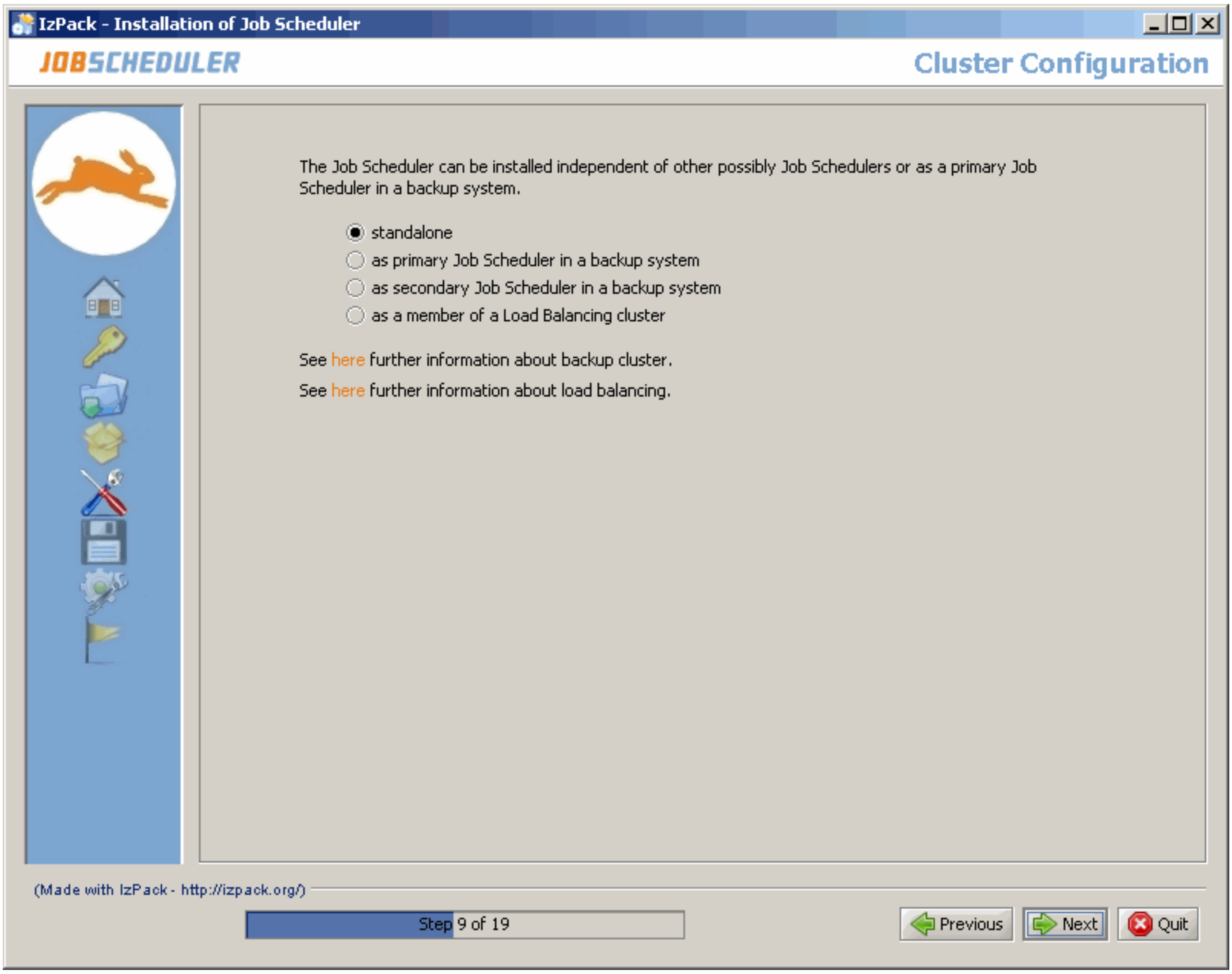
Enter the start time in the format HH:MM 23 : 45

□ with automatic download

A *Job Scheduler* 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 the *Job Scheduler* [appdata_path] with the operating system dependent name of

The *[appdata_path]/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 *[install_path]/doc/en/scheduler_update_service.pdf* documentation.

1.6.3 The Database Support Package Forms



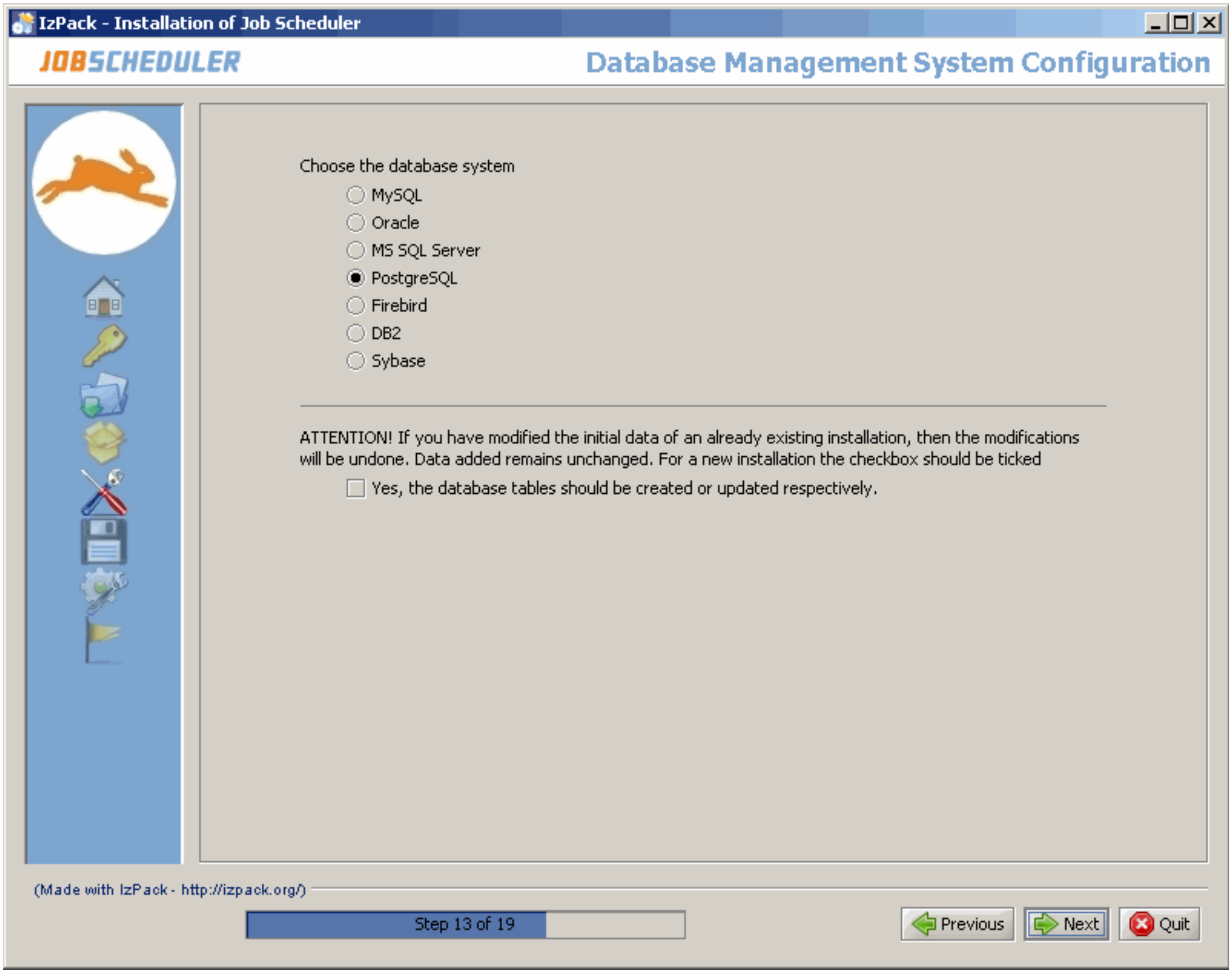
The Job Scheduler can be installed independent of other possibly Job Schedulers or as a primary Job Scheduler in a backup system.

● standalone
○ as primary Job Scheduler in a backup system
○ as secondary Job Scheduler 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.

The radio buttons in the form shown above determine whether the *Job Scheduler* should be installed "stand-alone" or in a backup system or a load balancing cluster. (see also Installation of a Backup System (page 32)).



IzPack - Installation of Job Scheduler
JOBSCHEDULER
Database Management System Configuration

Choose the database system

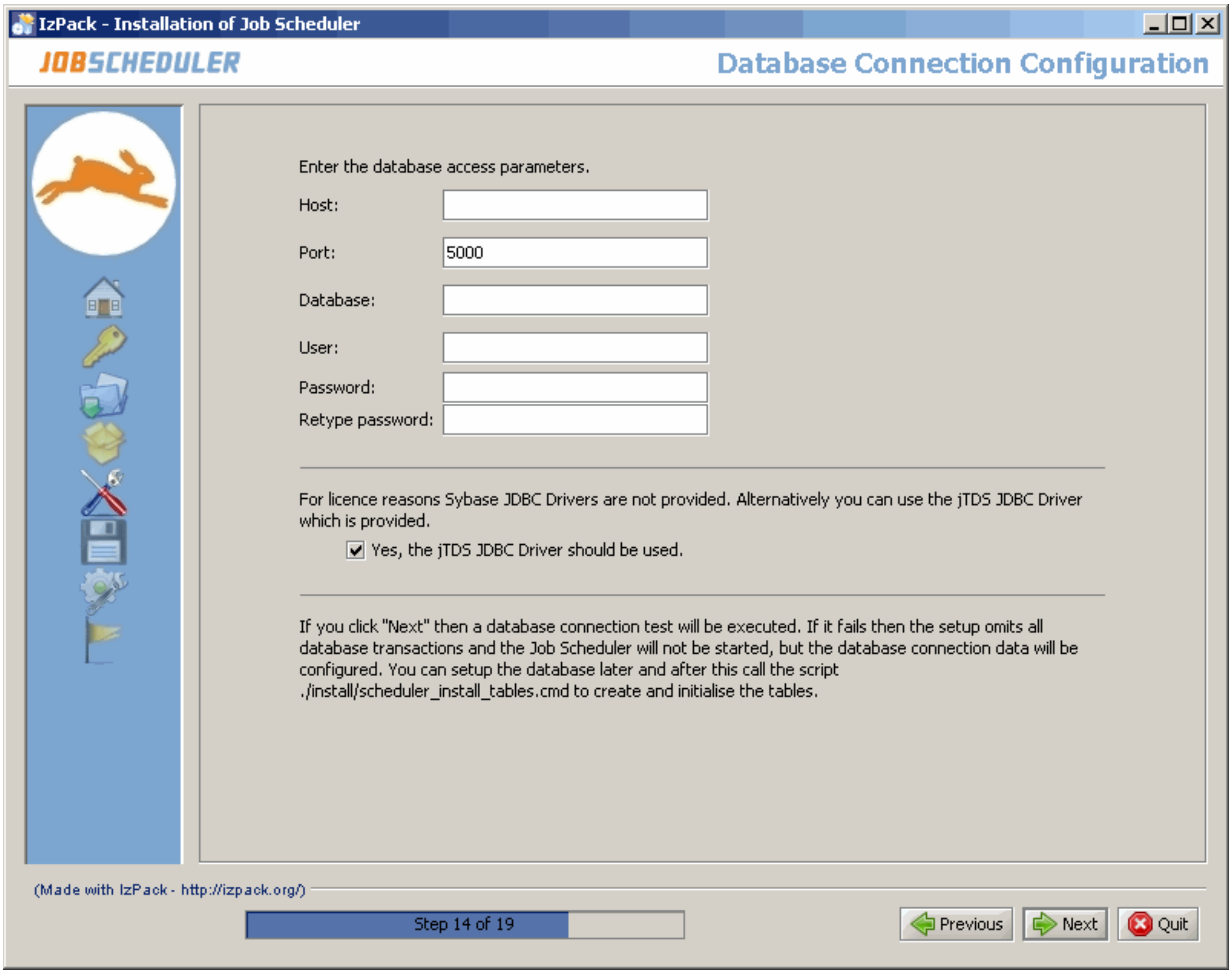
- ○ MySQL
- ○ Oracle
- ○ MS SQL Server
- ● PostgreSQL
- ○ Firebird
- ○ DB2
- ○ Sybase

ATTENTION! If you have modified the initial data of an already existing installation, then the modifications will be undone. Data added remains unchanged. For a new installation the checkbox should be ticked

□ 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 centre 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 23). If you have already installed another *Job Scheduler* with the same database connection then abandon this option.


IzPack - Installation of Job Scheduler
JOBSCHEDULER
Database Connection Configuration
(Made with IzPack - http://izpack.org/)
Step 14 of 19
Previous
Next
Quit

Enter the database access parameters.

Host:

Port: 5000

Database:

User:

Password:

Retype password:

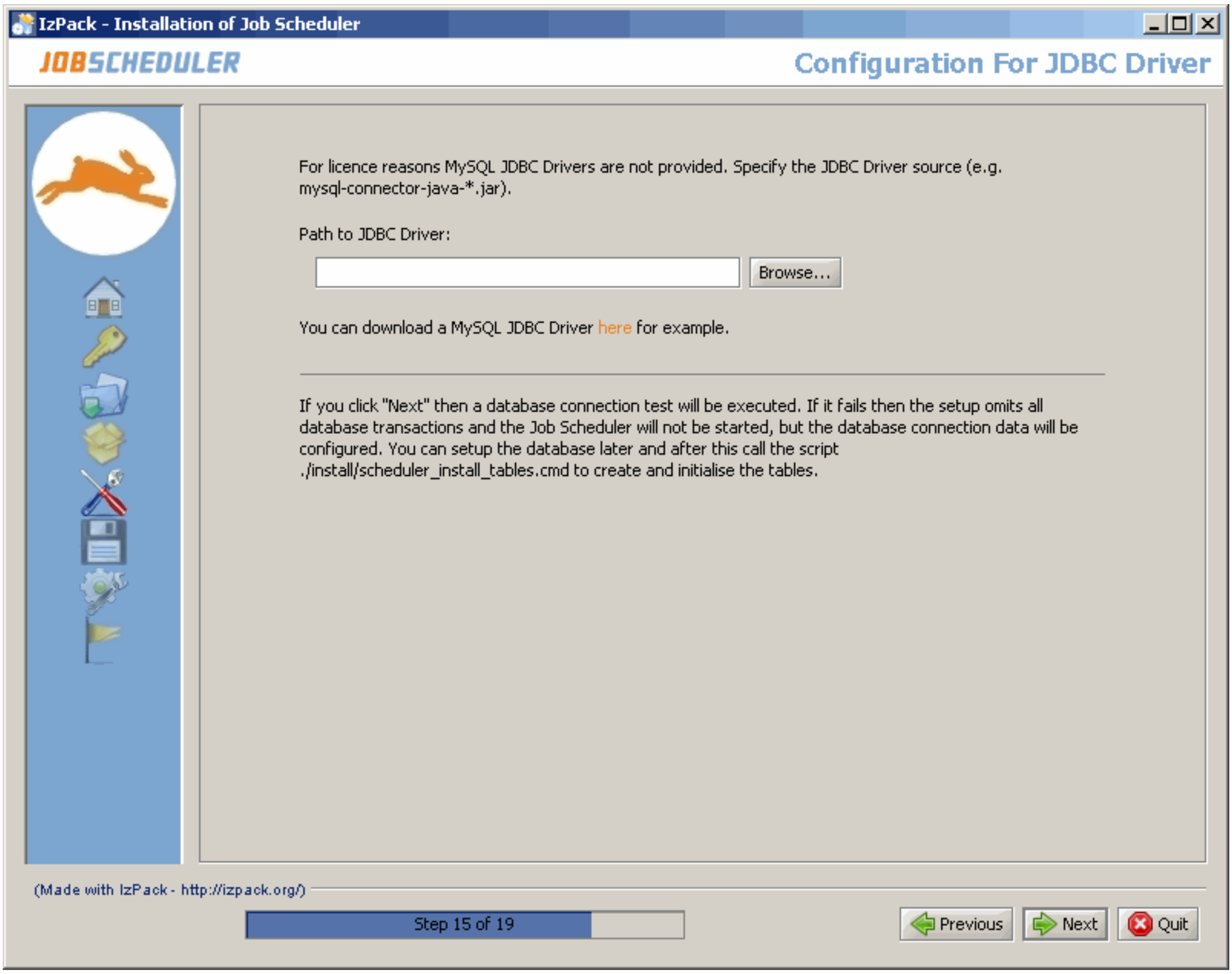
For licence reasons Sybase JDBC Drivers are not provided. Alternatively you can use the jTDS JDBC Driver which is provided.

☑ Yes, the jTDS JDBC Driver should be used.

If you click "Next" then a database connection test will be executed. If it fails then the setup omits all database transactions and the Job Scheduler 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.cmd to create and initialise the tables.

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-Treiber unchecked then you must enter your own JDBC driver in a next dialog.

This configuration is saved in the *[appdata_path]/config/factory.ini*, *[install_path]/web/custom/custom.inc.php* and *[appdata_path]/config/sos_settings.ini* files. All files can be changed manually (page 35) if required.


IzPack - Installation of Job Scheduler
JOBSCHEDULER
Configuration For JDBC Driver

For licence reasons MySQL JDBC Drivers are not provided. Specify the JDBC Driver source (e.g. mysql-connector-java-*.jar).

Path to JDBC Driver:

You can download a MySQL JDBC Driver here for example.

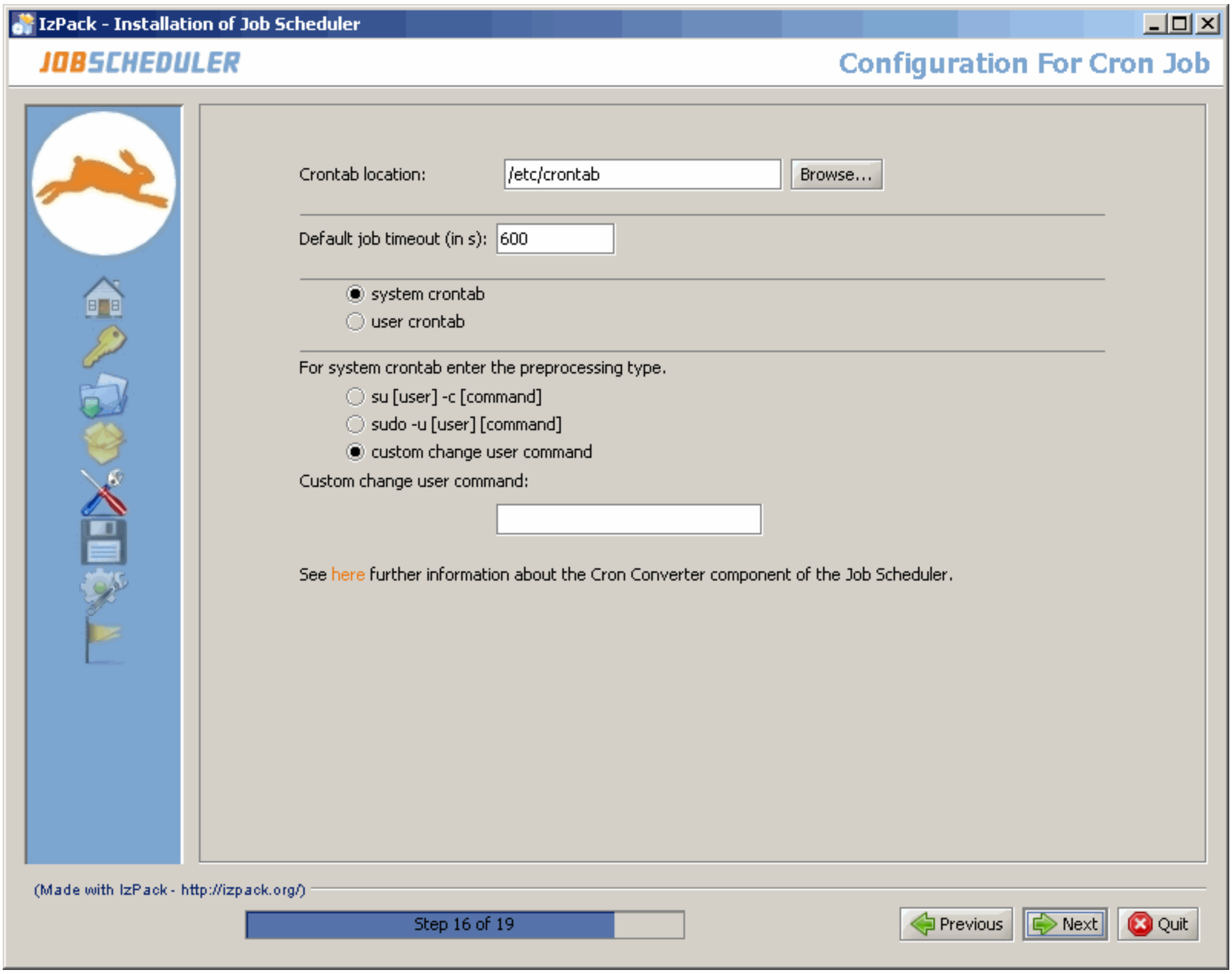
If you click "Next" then a database connection test will be executed. If it fails then the setup omits all database transactions and the Job Scheduler will not be started, but the database connection data will be configured. You can setup the database later and after this call the script ./install/scheduler_install_tables.cmd to create and initialise the tables.

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 Job Scheduler setup can be used when the appropriate checkbox in the previous form is activated.

As this driver will also be required by the Job Scheduler later on, it is copied by the setup into the *[install_path]/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 Packet Form


IzPack - Installation of Job Scheduler
JOBSCHEDULER
Configuration For Cron Job
(Made with IzPack - http://izpack.org/)
Step 16 of 19
Previous
Next
Quit
Browse...

Crontab location: /etc/crontab

Default job timeout (in s): 600

● 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 Job Scheduler.

This dialog is only available for Unix.

1.7 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.

You find in *[install_path]/Uninstaller/install.log* a precise list of all copied files via the setup.

The following directory structure should be found in the Job Scheduler [install_path]:

+ **bin** (Windows)

- **hostjava.dll** Program library
- **hostole.dll** Program library
- **jobeditor.cmd** Start script for the Job Configuration Editor
- **jobscheduler.cmd** Start script for the Job Scheduler

  - **jobscheduler_environment_variables.cmd** Environment variables of the Job Scheduler
  - **jobscheduler_event.cmd** Event handling script
  - **jobscheduler_client.pl** Perl script (TCP/UDP client for sending XML commands to a Job Scheduler)
  - **managedJobChainExport.cmd** Export script for Managed Jobs
  - **managedJobChainImport.cmd** Import script for Managed Jobs
  - **scheduler.exe** *Job Scheduler* executable file
  - **scheduler.exe.local** Dummy file for local usage of DLLs
  - **settingsImport.cmd** Import script for database settings (Managed Jobs)
  - **spidermonkey.dll** JavaScript (Mozilla) program library

+ **bin** (Unix)
  - **jobeditor.sh** Start script for the Job Configuration Editor
  - **jobscheduler.sh** Start script for the Job Scheduler
  - **jobscheduler_environment_variables.sh** Environment variables of the Job Scheduler
  - **jobscheduler_event.sh** Event Handling start script
  - **jobscheduler_client.pl** Perl script (TCP/UDP client for sending XML commands to a Job Scheduler)
  - **managedJobChainExport.sh** Export script for Managed Jobs
  - **managedJobChainImport.sh** Import script for Managed Jobs
  - **scheduler** *Job Scheduler* executable binary file
  - **scheduler_safe.sh** Watchdog script to respawn the Job Scheduler
  - **settingsImport.sh** Import script for database settings
  - **setuid** Program to process scripts in a different user context, see FAQ
+ **db** Database objects
  - **settings_insert_en.xml** (Managed Jobs)
  - **settings_insert_de.xml** (Managed Jobs)
  + **mssql** MS SQL Server 2000, 2005
    - **scheduler.sql**
    - **scheduler_sanity.sql**
    - **scheduler_sanity_insert.sql**
    - **acl.sql** (Managed Jobs)
    - **acl_insert.sql** (Managed Jobs)
    - **mails.sql** (Managed Jobs)
    - **scheduler_managed.sql** (Managed Jobs)
    - **scheduler_managed_insert.sql** (Managed Jobs)
    - **settings.sql** (Managed Jobs)
    - **settings_insert.sql** (Managed Jobs)
    - **user_attributes.sql** (Managed Jobs)
    - **user_groups.sql** (Managed Jobs)
    - **user_groups_insert.sql** (Managed Jobs)
    - **user_variables.sql** (Managed Jobs)
    - **user_variables_insert.sql** (Managed Jobs)
    - **users.sql** (Managed Jobs)
    - **users_insert.sql** (Managed Jobs)
  + **mysql** MySQL 4.1, 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)
        - **sos.sql** (Managed Jobs)
    + **db2** IBM DB2 8
        - ***.sql** (see pgsql directory)
    + **sybase** Sybase 15.x
        - ***.sql** (see mssql directory)

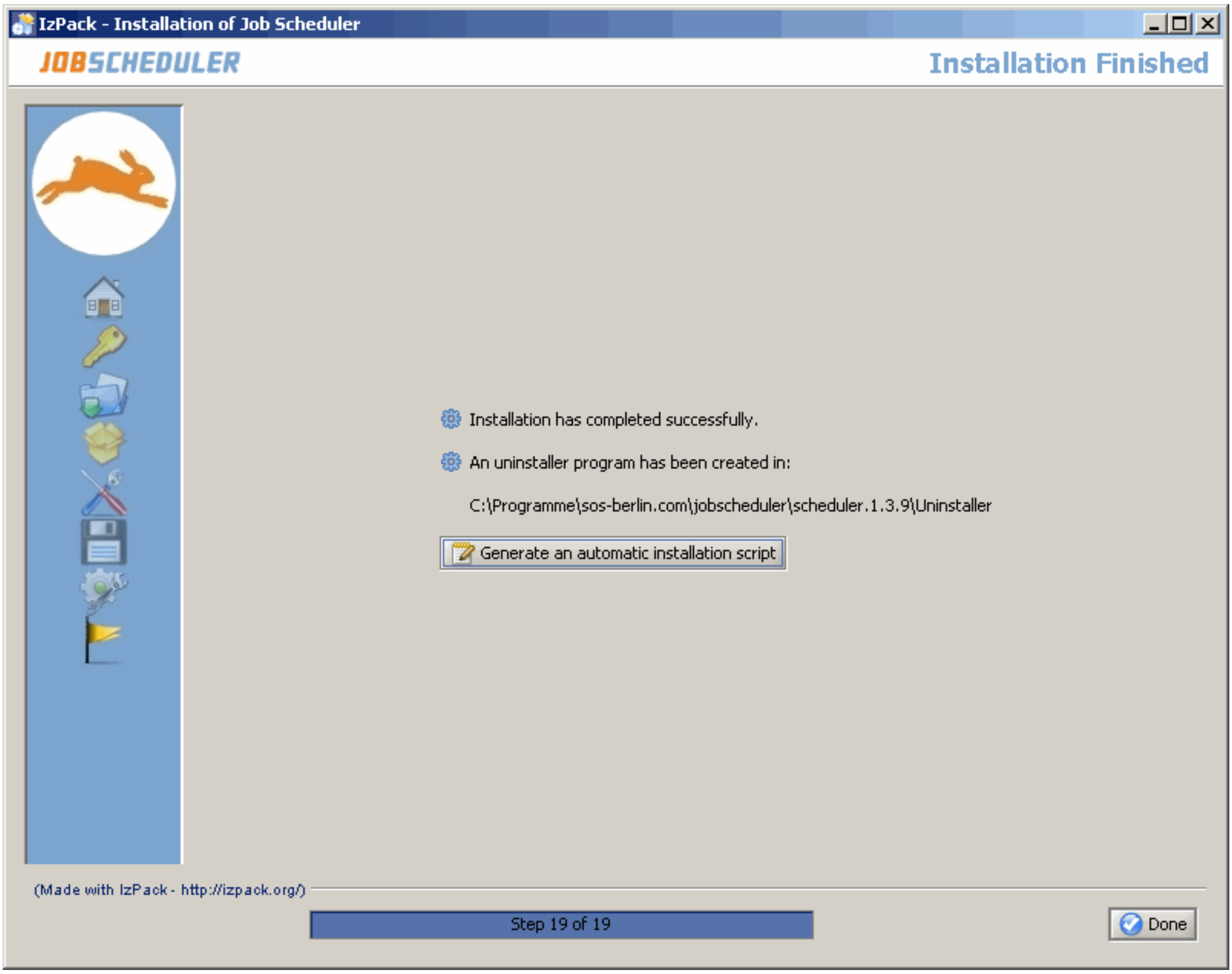
+ **doc** Documentation including API and Tutorial
+ **install** (Windows)
+ **lib**
    - ***.jar** Java archives (for Java jobs)
    - **scheduler.dll** for Java debugging (Windows)
    - ***.so, *.sl, *.a** libraries (Unix)
+ **operations_gui** HTML and Javascript files of the OperationsGUI
+ **Uninstaller** Program to uninstall the Job Scheduler
+ **web** PHP interface (Web Interface)
    + **custom** Configuration file for the PHP interface
    + **doc** Documentation available via the web server
    + **...** further directories

The following directory structure should be found in the Job Scheduler [appdata_path]:

+ **config**
    + **cache** Configuration directory in a *Workload Scheduler* (Replicate of a *Supervisor* remote directory)
    + **events** (Housekeeping jobs/event handling)
    + **live** Local configuration directory for the *Job Scheduler* (Hot Folder)
        + **sos**
            + **events** (Event Handler Jobs)
            + **housekeeping** (Housekeeping Jobs)
            + **managed** (Managed Jobs)
            + **mysql** (MySQL Maintenance Jobs)
            + **update** (Update Service Job)
    + **operations_gui**
        - **custom.js** Konfiguration-Datei der OperationsGUI
    + **remote** Local configuration directory for a *Workload Scheduler* on a *Supervisor*
    - **factory.ini** Runtime configuration file
    - **scheduler.xml** The *Job Scheduler* XML configuration file
    - **scheduler.xsd** XML configuration files schema definition
    - **scheduler_mail.xsl** style sheet for emails with log files
    - **sos.ini** licence file
    - **sos_settings.ini** Database connection
    - **default.xslt** (Managed Jobs)
    - **mail.xslt** (Managed Jobs)
    - **factory_mysql.ini** (MySQL Maintenance Jobs)
    - **replication_master_settings.ini** (MySQL Maintenance Jobs)
    - **replication_slave_settings.ini** (MySQL Maintenance Jobs)
+ **jobs** XML documentations miscellaneous standard jobs
+ **logs** Depository for log files

1.8 Automatic Installation

After the *Job Scheduler* setup has been completed, a form appears for saving an XML script file. This script can be later used for automatic installation of the Job Scheduler. All the variables entered during setup are then saved in this file. A separate form for generating and saving this file is opened by clicking on the *Generate Automatic Installation Script* button. This automation script can then be used to ease the repeated installation of the Job Scheduler on different computers.


IzPack - Installation of Job Scheduler
JOBSCHEDULER
Installation Finished
(Made with IzPack - http://izpack.org/)
Step 19 of 19
Done

Installation has completed successfully.

An uninstaller program has been created in:

C:\Programme\sos-berlin.com\jobscheduler\scheduler.1.3.9\Uninstaller

Generate an automatic installation script

Instructions for starting the automatic setup script can be found under Batch Installation (page 24).

1.9 Database Configuration

It is recommended that the Job Scheduler 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. MS SQL Server, MySQL, PostgreSQL, DB2, Firebird, Sybase and Oracle database systems are supported. The Job

Scheduler setup program creates the necessary database tables if the *Database Support (page 8)* package is installed and the database connection is specified in the appropriate setup form.

The database configuration information is saved in the *[appdata_path]/config/factory.ini* and *[install_path]/web/custom/custom.inc.php* files.

1.9.1 Manual Creation of Database Table

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

Ensure that the database connection is correctly entered in the *[appdata_path]/config/factory.ini*, *[appdata_path]/config/sos_settings.ini* and *[install_path]/web/custom/custom.inc.php* configuration files (page 35).

1.10 Starting and Stopping the Job Scheduler

1.10.1 Job Scheduler Demon on Unix

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

```
[install_path]/bin/jobscheduler.sh start
```

```
[install_path]/bin/jobscheduler.sh stop
```

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

If you want the Job Scheduler to be started automatically at server startup, then please copy this script and the `[install_path]/bin/jobscheduler_environment_variables.sh` file to the appropriate startup/shutdown directory - usually this is `/etc/init.d`.

The Job Scheduler is not automatically started after installation - the above script should be used for this.

1.10.2 Job Scheduler Service for Windows

On Windows systems, the Job Scheduler is installed as service. You can find the Job Scheduler service by opening the Windows service panel and looking for a service with a name starting with "SOS Job Scheduler".

To start the Job Scheduler manually, ensure that the service has not already been started and use the following script:

```
[install_path]/bin/jobscheduler.cmd start
```

```
[install_path]/bin/jobscheduler.cmd stop
```

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

The *Job Scheduler* service is automatically started after the installation.

2 Batch Installation

Note that when the *Job Scheduler* installation is started from a parameterized XML file, no dialog forms will appear.

```
shell>java -jar [setup.jar] [batch_install.xml]
```

Note also that [setup.jar] is the Setup program (page 4) for the operating system being used and [batch_install.xml] is a specific XML file (see below). Such XML files are generated, for example, by clicking on the *Generate Automatic Installation Script* button of the setup dialog.

A parameterized XML installation file must have the following form:

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

The Job Scheduler is available with a dual licensing model.
- GNU GPL 2.0 License (see http://www.gnu.org/licenses/gpl-2.0.html)
- Job Scheduler 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 Scheduler 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>[:choose absolute installation path:]</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 Scheduler 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 Job Scheduler
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: Job Scheduler
            Job Scheduler 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="true"/>

       <!-- 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: Web Interface
            The Web Interface enables the monitoring Job Schedulers and control of
            jobs. The Job Scheduler package above includes an operationGui for
            monitoring too, so that this Web interface is not essential.
            The installation of PHP not below version 4.3 is required.
            Database support is required to operate this feature.-->
       <pack index="3" name="Web" selected="false"/>

       <!-- Package: Managed Jobs
            Managed Jobs are stored in a database and automatically distributed to
            one or more Job Schedulers. Database support and Web package is
            required to operate this feature. -->
       <pack index="4" name="Managed Jobs" selected="false"/>

       <!-- 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 Job Scheduler. -->
       <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 Job Scheduler with a
         crontab file. For that purpose, the Job reads the crontab file and
         dynamically adjusts the Job Scheduler 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 Job Scheduler
             is operated. Please avoid 'localhost' as host. Sometimes that will
             lead to problems. -->
        <entry key="serviceHost" value=""/>

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

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

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

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

        <!-- The Job Scheduler can be installed independent of other possibly
             Job Schedulers, as a primary Job Scheduler in a backup system or
             as a backup Job Scheduler. Use '' for a standalone, '-exclusive'
             for a primary or '-exclusive -backup' for a backup Job Scheduler.
             A database is required for a backup system. All Job Schedulers in
             a backup system must have the same Job Scheduler ID and the same
             database. Further you can set '-distributed-orders' for a load
             balancing cluster. -->
        <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 Job Scheduler
             as event handler. Choose this option if you intend to use Job
             Scheduler Events and
             - this Job Scheduler instance is the only instance which processes
               Events
             - this Job Scheduler instance is a supervisor for other
               Job Schedulers 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 Job Scheduler 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
                 Job Scheduler 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=""/>

            <!-- You can choose between 'on' or 'off' to (re)create the database
                 tables. ATTENTION! If you have modified the initial data of an
                 already existing installation, then the modifications will be
                 undone. Data added remains unchanged. For a new installation the
                 value should be 'on'. -->
            <entry key="databaseCreate" value="off"/>

        </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, postgreSQL 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
                 -->
            <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>
    <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>
```

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

A sample XML file with the name *scheduler_install.xml* is contained in the installation archive.

3 Multiple Installation

3.1 Reinstallation of the Job Scheduler

Reinstallation means an installation in the same directory on the same computer as an existing installation of the Job Scheduler.

For this the setup is only pratical with reservation, because if you want to update a *Job Scheduler* 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 also provided Update-Setup.

3.2 Installation Alongside an Already Existing Installation

A parallel installation is defined as an installation of the Job Scheduler on the same computer as an existing installation, but in a new directory.

The following points must be observed when completing the *Network Configuration* (page 11) form of the *Job Scheduler* basic package setup:

- The *Scheduler ID* must be unique amongst all the Job Schedulers installed on one computer.
  On Windows the Job Scheduler ID is used after the setup is completed to set the name of the Job Scheduler service in the *sos_scheduler_[scheduler_id]* form.
- The *TCP port* must also be unique amongst all the Job Schedulers installed on one computer.

It is recommended that all Job Schedulers installed on a computer or in a network use the same database connection. This *must* be the case when the *Managed Jobs* package, a backup cluster or Load Balancing is to be used.

The *Web Interface* package does not need to be reinstalled, as long as the database connection for the new installation remains unchanged. Instead, it is recommended that a *Supervisor* is defined in the *[appdata_path]/config/scheduler.xml* (page 35) file. For further information about look *Supervisor* at the *Job Scheduler* documentation.

4 The Installation of a Backup Cluster

A Backup Cluster is installed by first of all installing a primary *Job Scheduler* for the cluster as described in the Installation (page 4) chapter of this documentation. In the course of setting up this Job Scheduler, the *Database Support* package should be selected, as the operation of a backup cluster can only take place with database support. In the *Backup Configuration* dialog, the installation as a primary backup cluster *Job Scheduler* should be selected. In the last part of the installation dialog, the automatic installation script *must* be generated.

The installation of the backup *Job Schedulers* is then carried out as described under Batch-Installation (page 24). Before this is done, however, the installation script which was created whilst installing the primary *Job Scheduler* should be opened in a text editor. The settings for the '<installpath>' element and the 'value' attribute of the <entry> element are then modified using 'key="clusterOptions"'.

The 'clusterOptions' is then given the values '-exclusive -backup -backup-precedence=[N]', where [N] is an integer.

When more than one backup *Job Scheduler*s are able to replace the failed primary *Job Scheduler* (-exclusive), then the operation will be taken over by the *Job Scheduler* with the lowest -backup-precedence value. Should the -backup-precedence=[N] not be specified, then an initial value of '1'will be allocated (0 is reserved for the primary *Job Scheduler*). The value of the 'value' atribute in the <entry>-Elements with 'key="databaseCreate"' should be set to 'off', as the database has already been created when the primary *Job Scheduler* was set up.

Similarly, other values such as 'serviceHost' must be modified. On the other hand, the database and Scheduler ID settings may not be altered, as all *Job Schedulers* in a backup cluster must have the same database connection and the same Scheduler ID.

On Windows systems, a service will be created for the backup *Job Scheduler* with the name *sos_scheduler_[scheduler_id]_backup*.

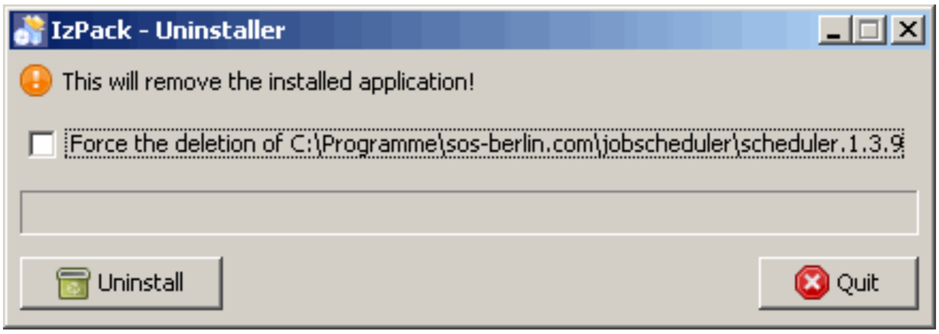
5 Deinstallation

5.1 Removal Using the Uninstaller

The Uninstaller *[install_path]/Uninstaller/uninstaller.jar* is initialized by the setup program used to install the Job Scheduler. The Uninstaller is started using:

```
windows-shell>java -jar [install_path]\Uninstaller\uninstaller.jar
unix-shell>java -jar [install_path]/Uninstaller/uninstaller.jar
```

which opens a dialog box asking for the removal of the Job Scheduler to be confirmed.



This will remove the installed application!

□ Force the deletion of C:\Programme\sos-berlin.com\jobscheduler\scheduler.1.3.9

A database created for the Job Scheduler must be deleted manually. Similarly, any virtual directories created on the web server must be deleted manually as well.

For Unix Users
The Uninstaller is a dialog program which requires that an X-Server is installed on the client computer.

For Windows Users
The uninstall program may be started using a double click when "jar" files are linked to the file

```
"[Path to Java installation JRE]\bin\javaw.exe" -jar "%1" %*
```

When an IIS web server is configured for the *Job Scheduler* web interface, then the relevant virtual directories are to be deleted before removal of the Job Scheduler, otherwise the associated physical directories will not be completely removed by the Uninstaller.

The "SOS Job Scheduler id=[scheduler_id]" service should be removed manually after uninstalling a Job Scheduler. It is important to note here the correct [scheduler_id] - that is the ID specified during installation of the Job Scheduler. 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:

```
C:\>net start sos_scheduler_[scheduler_id]
```

on the command line. Depending on the status of the service, a message similar to one of the following statements will appear:
The requested service cannot be started. It has either been deactivated or is not associated with an activated device.
or
The name of the service is invalid.
Should the service only have been deactivated, then a renewed installation of a Job Scheduler with the same [scheduler_id] will only be possible after the computer has been restarted.

5.2 Manual Removal on Windows

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

- **Reconfigure the Web Server**
  Should a web server have been configured for the Job Scheduler web interface, then it is necessary to remove the associated virtual directories. This is particularly important when IIS is used as otherwise it will not be possible to completely remove all directories.

- **Stop the Job Scheduler**
  ```
  C:\>[install_path]\bin\jobscheduler.cmd stop
  ```
  An error message will be shown, should the Job Scheduler already have been stopped. This message can be ignored.

- **Remove the Job Scheduler Service**
  ```
  C:\>[install_path]\bin\jobscheduler.cmd remove
  ```

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

- **Deregister the hostole.dll program library**
  ```
  C:\>regsvr32 /u [install_path]\bin\hostole.dll
  ```

- **Delete all files and directories**
  ```
  C:\>rmdir /S /Q [install_path]
  C:\>rmdir /S /Q [appdata_path]
  ```

5.3 Manual Removal on Unix

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

- **Reconfigure the Web Server**
  Should a web server have been configured for the *Job Scheduler* web interface, then the corresponding virtual directories should be removed.

- **Stop the Job Scheduler**
  ```
  shell>[install_path]/bin/jobscheduler.sh stop
  ```
  An error message will be shown, should the Job Scheduler already have been stopped. This message can be ignored.

- **Remove the Database**
  The documentation for any database which may have been installed for the Job Scheduler should be consulted for instructions as to its removal.

- **Delete all Files and Directories**
  ```
  shell>rm -r -f [install_path]
  shell>rm -r -f [appdata_path]
  ```

6 Configuration

The *Job Scheduler* is configured using the following files:

- **factory.ini**
- **scheduler.xml**
- **custom.js** (konfiguriert die OperationsGUI)
- **custom.inc.php** (configures the web interface)
- **jobscheduler_environment_variables.sh**

These files are configured during the Job Scheduler setup, using the information entered at the time.

6.1 The factory.ini File

The *factory.ini* file is to be found in the *[appdata_path]/config* directory. E-mail settings, information about the database connection and the Java archives classpath are saved in this file. Further details about the entries in this file are to be found in the Job Scheduler documentation.

6.2 The scheduler.xml and scheduler.xsd Files

The *scheduler.xml* and *scheduler.xsd* files are to be found in the *[appdata_path]/config* directory. The *Job Scheduler* port information are to be found in the *scheduler.xml* file. Further details about this file are to be found in the *Job Scheduler* documentation.

The *scheduler.xml* file is validated with the *scheduler.xsd* schema file, which contains the configuration XML configuration.

A graphical editor (JOE) is delivered with the *Job Scheduler* for administering the job configurations specified in the *scheduler.xml* file.
On Windows systems this editor is started as follows:

```
[install_path]\bin\jobeditor.cmd
```

An X-Server is necessary on Unix systems and the editor is started using the script:

```
[install_path]/bin/jobeditor.sh
```

The Job Editor automatically validates configuration files against the *scheduler.xsd* schema.

6.3 The custom.js File

The *custom.js* file is to be found in the *[appdata_path]/operations_gui* directory. Beside other settings you can configure the language and filters in particular.

6.4 The jobscheduler_environment_variables.sh File

The *jobscheduler_environment_variables.sh* file is particularly relevant for Unix systems and is found in the *[install_path]/bin* directory. In this file the *LD_LIBRARY_PATH* is set, which must be customized, if the *Job Scheduler* should not find the java environment.
In this case the following error is logged in *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]]
```

The above error can also be logged on AIX systems, even though the *LD_LIBRARY_PATH* is correct. This happens when the Java installation only provides a *libjvm.a* instead of *libjvm.so* library. In this situation, a symlink should be created:

```
sh> cd $JAVA_HOME/bin/j9vm
sh> ln -s libjvm.a libjvm.so
```

If you modify the *Scheduler ID* on windows then note that the corresponding service must be reinstalled.

6.5 The custom.inc.php File

The *custom.inc.php* is to be found in the *[install_path]/web/custom* directory, if the *Web Interface (page 8)* package was installed during setup. This file is used to specify database connection information; the language used in the *Job Scheduler*, host and port information as well as the timeout value for TCP commands.

Language
English and German are supported. The PHP constant *SOS_LANG* is used to specify the language used. This constant takes a two letter country code (written lower case). Should no entry be made here, then German will be used.

- For English:
  ```
  if(!defined('SOS_LANG')) { define ( 'SOS_LANG', 'en' ); }
  ```

- For German:
  ```
  if(!defined('SOS_LANG')) { define ( 'SOS_LANG', 'de' ); }
  ```

Database Connection
The PHP constant *APP_CONNECTION_AUTH* is used to set the database connection in the form:

```
if(!defined('APP_CONNECTION_AUTH')) { define ( 'APP_CONNECTION_AUTH',
 '-db=[databasename] -user=[username] -password=[password]
  -host=[servername oder -IP]:[port]' ); }
```

Should a value for '[port]' not have been set, then the standard port used by the database will be used. Should a value for '-host' not have been set, then 'localhost' and the standard port will be used.

Database Type
Oracle, MySQL, Microsoft SQL Server, PostgreSQL, DB2, Firebird, Sybase and ODBC data sources are supported. The database type is set using the *APP_CONNECTION_CLASS* PHP constant as follows.

- For Oracle:
  ```
  if(!defined('APP_CONNECTION_CLASS')) {
     define ( 'APP_CONNECTION_CLASS', 'sos_oracle_record_connection' ); }
  ```

- For MySQL:
  ```
  if(!defined('APP_CONNECTION_CLASS')) {
     define ( 'APP_CONNECTION_CLASS', 'sos_mysql_record_connection' ); }
  ```

- For Microsoft SQL Server:
  ```
  if(!defined('APP_CONNECTION_CLASS')) {
     define ( 'APP_CONNECTION_CLASS', 'sos_mssql_record_connection' ); }
  ```

- For PostgreSQL:
  ```
  if(!defined('APP_CONNECTION_CLASS')) {
     define ( 'APP_CONNECTION_CLASS', 'sos_pgsql_record_connection' ); }
  ```

- For DB2:
  ```
  if(!defined('APP_CONNECTION_CLASS')) {
     define ( 'APP_CONNECTION_CLASS', 'sos_db2_record_connection' ); }
  ```

- For Firebird:
  ```
  if(!defined('APP_CONNECTION_CLASS')) {
     define ( 'APP_CONNECTION_CLASS', 'sos_fbsql_record_connection' ); }
  ```

- For Sybase:
  ```
  if(!defined('APP_CONNECTION_CLASS')) {
     define ( 'APP_CONNECTION_CLASS', 'sos_sybase_record_connection' ); }
  ```

- For ODBC Data Sources:
  ```
  if(!defined('APP_CONNECTION_CLASS')) {
     define ( 'APP_CONNECTION_CLASS', 'sos_odbc_record_connection' ); }
  ```

The Monitoring Job Scheduler Host

```
if(!defined('APP_SCHEDULER_HOST')) { define ( 'APP_SCHEDULER_HOST', 'localhost' ); }
```

The Monitoring Job Scheduler TCP Port

```
if(!defined('APP_SCHEDULER_PORT')) { define ( 'APP_SCHEDULER_PORT', '4444' ); }
```

The Monitoring Job Scheduler ID

```
if(!defined('APP_SCHEDULER_ID')) { define ( 'APP_SCHEDULER_ID', 'scheduler' ); }
```

Timeout
The web interface sends commands to the Job Scheduler using TCP. Should these commands not be answered in the time specified here (in seconds), then the web interface terminates the TCP connection.

```
if(!defined('APP_SCHEDULER_TIMEOUT')) { define ( 'APP_SCHEDULER_TIMEOUT', '5' ); }
```

6.6 Configuration of the Web Server

Selection of the *Web Interface (page 8)* package during setup requires a web server that is configured for the use of PHP in version 4.3 or higher. This server should be configured so that the directory *[install_path]/web* is available. Further details about the creation of virtual directories can be found in the web server documentation. Furthermore the web server requires read and write permissions in *[install_path]/web* and *[install_path]/web/custom*.

Example - for Apache (*httpd.conf*):
```
Alias /scheduler/ [install_path]/web/
```

The following modules must be activated in the `php.ini` PHP configuration file:

- php_domxml (already implemented - depends on the PHP version)
  The PHP5 dom extension and the PHP4 domxml extension must not be loaded at the same time.
- php_oci8 (when an Oracle database is used)
- php_pgsql (when an PostgreSQL database is used)

- php_mssql (when a MS SQL Server database is used)
- php_mysql (when a MySQL database is used - depends on the PHP version already implemented)
- php_ibm_db2 (when a DB2 database is used)
- php_interbase (when a Firebird database is used)
- php_sybase_ct (when a Sybase database is used)

More settings are mandatory in the `php.ini` PHP configuration file:

- `session.use_cookies = 1`
- `session.auto_start = 0`
- `session.use_trans_sid = 1`
- `file_uploads = On` (if the import function of the web interface will be used)

The web server should be restarted after changes are made to the `php.ini` file.

7 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 Job Scheduler. Should such a release be available, then this information will be conveyed to the query initiator.

A job is delivered with the Job Scheduler which asks this web service once a week if a newer version of the Job Scheduler 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.

7.1 SchedulerUpdate Web Service

The service is available 24 hours a day:
- it accepts "CheckRequest" queries
- it determines whether a new version is available
- it sends a reply to originator of the query

The Query Structure:

```
<CheckForUpdateRequest>
 <hostname>Client hostname</hostname>
 <release>Release nr. from [appdata_path]/config/.version</release>
 <os>Operating system from Java System request</os>
 <os_install>Operating system from [appdata_path]/config/.version </os_install>
 <product>scheduler</product>
 <automatic_download>[0|1]</automatic_download>
</CheckForUpdateRequest>
```

The Response Structure:

```
<CheckForUpdateAnswer>
 <release>Value from request</release>
 <new_release>Current release nr. available</new_release>
 <os>Value from request</os>
 <os_install>Value from request</os_install>
 <automatic_download>Value from request</automatic_download>
 <update_needed>1, should an update be available</update_needed>
</CheckForUpdateAnswer>
```

7.2 CheckForUpdate Job (Client)

Method of Operation:
- the job starts once a week (this interval is preset)
- it sends a synchronous CheckRequest to the WebService
- details about the current release are taken from *[appdata_path]/config/.version*
- the e-mail address to which the notification is to be sent is read from the *[appdata_path]/config/factory.ini* file
- the connection timeout = 30 seconds

Job Definition

The implementation can be found in the sos.scheduler.jar archive.

```
<job name="check_for_update" title="Automatic Update Procedure for the Job
Scheduler">
 <script java_class="sos.scheduler.job.JobSchedulerCheckUpdates" language="java"/>
 <run_time let_run="no">
   <weekdays>
     <day day="1">
       <period single_start="20:00"/>
     </day>
   </weekdays>
 </run_time>
</job>
```

The config/.version File Structure

```
[scheduler]
release=x.x.x.x
os_install=[windows|linux|solaris|hpux-ia64-32|aix]
```

The following order parameters can be specified:

```
webserviceUrl:           http://www.sos-berlin.com/check_for_update
product:                 scheduler
ftp_host:                www.sos-berlin.com
ftp_port:                21
ftp_user:                anonymous
ftp_password:
ftp_transfer_mode:       binary
ftp_passive_mode:        1
ftp_remote_dir
ftp_local_dir
ftp_automatic_download:  0
```

The job sends a query to the web service and waits for a response. The connection timeout is permanently set to 30 seconds.

Should the web service reply that a new release is available, then an appropriate e-mail is sent to the system administrator:

```
There is a new version of the job scheduler available.
Your version is: 1.2.3.3
New version is: 1.2.7
The update-file has been downloaded
to:/home/sos/scheduler/scheduler_linux_update.tar.gz
```

An appropriate FTP transfer will be started if the job has been configured for automatic downloading of updates.

In the event of an error, the job will add a warning to the log file.

The Request Structure

The name of the operating system is read from *System.getProperties()* and *"os.name"* and the name of the host from *InetAddress.getLocalHost* with *getHostName()*.

A typical query would be:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<wsa:To xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
http://localhost:4455/check_for_update</wsa:To>
<wsa:ReplyTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
<wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous<
/wsa:Address>
</wsa:ReplyTo>
</soapenv:Header>

<soapenv:Body>
 <addOrder xmlns="http://www.sos-berlin.com/scheduler">
   <jobchain>check_for_update</jobchain>
   <title>Scheduler: Check for Update</title>
   <xml_payload>
     <CheckForUpdateRequest>
       <hostname>myHost</hostname>
       <release>1.2.3.4</release>
       <os>Windows XP</os>
       <os_install>windows</os_install>
       <product>scheduler</ product >
       <automatic_download>1</automatic_download>
     </CheckForUpdateRequest>
   </xml_payload>
 </addOrder>
</soapenv:Body>
</soapenv:Envelope>
```

7.3 Handling Multiple Updates

Should a new release be found, then this will be noted in the *[appdata_path]/config/.version* file. After the update has been downloaded, the directory in which it is saved is also noted in this file. The *[appdata_path]/config/.version* file appears as shown below after a new release has been registered:

```
[scheduler]
release=1.2.3.8
new_release=1.2.9
os_install=windows
downloaded_file=filename
```

Should the new release not have been downloaded before the update check is rerun (i.e. one week later), then no further e-mails informing the administrator about a new release will be sent out. After the file has been successfully downloaded once, it will not be requested again.

Instead, the following will be written in the log file:

```
"You already have downloaded new release, but it has not been installed"
"You have already received an e-mail about a new release, but this new release has
not been installed.
See file filename"
```

8 Operation on 64bit Systems

You can operate the *Job Scheduler* in 64bit environments on the supported platforms. However, keep the following explanations and requirements in mind:

- The *Job Scheduler* is a 32bit program.
  The Job Scheduler is implemented as a 32bit executable. You may be asking yourself why this software has not been ported to 64bit? The answer is quite simple - it is not necessary.
  - You can run the 32bit Job Scheduler executable on every supported 64bit platform (Windows, Linux, Solaris, HP-UX Itanium, AIX). Because the Job Scheduler creates platform specific processes, its operation is independent of whether or not 64bit programs are to be scheduled by jobs on a 64bit platform.
  - The possible difference in execution speed between a 32bit and a 64bit Job Scheduler executable would be negligible. Most of the execution time is consumed by jobs and not by the Job Scheduler itself.
  - Modifying the Job Scheduler code for 64bit support would require that the code was forked. Currently, 32bit platform downloads (Windows) represent most of the Job Scheduler installations and have to be supported. We would have to spend additional developer resources on code maintenance for both versions over a longer period of time. This, in turn, would slow down the current roadmap. We estimate the benefit from new features to be higher than the benefits that could be gained from the simplified installation of a native 64bit Job Scheduler.
- Java 32bit JRE
  A Java 32bit JRE is required for executing the setup and operating the Job Scheduler.
  In our experience, it is a good idea to use a JRE provided by SUN, as other distributions, such as IBM Websphere, tend to be somewhat incomplete or have been modified for individual products to an extent that they are not sufficiently standards compatible.
  - Windows 64bit
    You should not include a newly installed 32bit JRE permanently in your *PATH* variable (or change an existing JDK *JAVA_HOME* variable), as this would affect other programs that rely on a 64bit JRE. Installation should be carried out using the following steps:
    1. Download and install the 32bit JRE.
    2. Open a console window and temporarily add the location of the newly installed JRE to your *PATH* using:
       ```
       C:\>SET PATH=C:\Program Files (x86)\Java\jre1.6.0_01\bin;%PATH%
       ```
    3. Unzip the Job Scheduler download archive to a temporary directory, e.g. `C:\temp\scheduler`.
    4. Run Java with the Job Scheduler setup from the temporary directory in the console window with:
       ```
       C:\>java -jar C:\temp\scheduler\scheduler_win32.jar
       ```
    5. Having completed the setup, add the location of the Java Virtual Machine to the Job Scheduler *[appdata_path]\config\sos.ini* configuration file:
       ```
       [java]
       vm = C:/Program Files (x86)/Java/jre1.6.0_01/bin/client/jvm.dll
       ```
  - Unix 64bit
    You have to download a 32bit JRE if this version is not available on your machine.
    1. Download and install the 32bit JRE.
    2. Temporarily add the location of the newly installed JRE to your *PATH*:
       ```
       PATH=$HOME/jdk1.6.0_01/jre/bin:$PATH
       ```
    3. Unzip the Job Scheduler download archive to a temporary directory, e.g. `/tmp/scheduler` using:
       ```
       md /tmp/scheduler
       cd /tmp/scheduler
       tar xzf scheduler_linux.1.3.1.tar.gz .
       ```
    4. Run Java with the Job Scheduler setup from the temporary directory with:
       ```
       sudo java -jar /tmp/scheduler/scheduler_linux32.jar
       ```

5. Having completed the setup, add the location of the Virtual Machine to the *LD_LIBRARY_PATH* environment variable in the `[install_path]/bin/jobscheduler_environment_variables.sh` script:
   `LD_LIBRARY_PATH=$SCHEDULER_HOME/lib:/usr/local/lib:$HOME/jdk1.6.0_01/jre/lib/i386:$HOME/jdk1.6.0_01/jre/lib/i386/client:$LD_LIBRARY_PATH`

9 Troubleshooting

In addition to the following information, further assistance in troubleshooting issues can be found in the FAQ on the Job Scheduler web site.

Choosing the Appropriate JDBC Driver
The JDBC driver must correspond to the version of your database system. Should database connection problems occur in the Job Scheduler, then the following error may be displayed in the log files:

```
Error SOCKET-61 ECONNREFUSED Connection refused (TCP-Port not available) [connect].
```

For legal reasons we cannot give any recommendations here but in our experience:

- MySQL
  `mysql-connector-java-3.1.8-bin.jar` works well with MySQL version 4.x.
- PostgreSQL
  we use the PostgreSQL 7 JDBC driver in the setup as version 8 currently does not support creating procedures via JDBC. However, we use version 8 of the JDBC driver for the Job Scheduler operation and we are not aware of major errors occurring with this driver version.
- SQL Server
  The `sqljdbc.jar` JDBC driver can be used with both 2000 and 2005 versions of SQL Server. However, an older JDBC driver version may not work with the 2005 SQL Server.
- Oracle
  we are not aware of any problems occurring with the Job Scheduler when using the ojdbc14.jar JDBC driver together with Oracle 9.2 and 10g.

ANSI Mode in MySQL
When connecting to a MySQL database, the Job Scheduler tries to switch to ANSI mode. This mode is essential for operation of our software as we have to support a variety of database systems and use ANSI compliant SQL. However, automatic switching to ANSI mode does not work with older versions of MySQL 4.0.x. You have to set this mode manually in the database server. Open the `my.cnf` file and insert the `SQL_MODE=ANSI_QUOTES` entry in the `[mysqld]` section. Alternatively, add the `--ansi` parameter to your MySQL start script. You have to restart the database server after making this change to the configuration.

Lost MySQL Database Connections
You may encounter the error:

```
Error connecting to [host]:[port]: SOS-JAVA-105 Java-Exception
java.sql.SQLException("No operations allowed after connection closed."),
methode=rollback []
```

If your MySQL database connection has been idle for some hours without any jobs being run, then MySQL will close the connection without informing the client - in this case, without informing the Job Scheduler. To change this behaviour you can change the value of the `wait timeout` system variable. This value defines the maximum duration of non-interactive idle connections to the database in seconds.
Alternatively, you could run a job such as `scheduler_dequeue_mail` that is often repeated: this job dequeues mails that have previously been stored in case of a failure in the mail server and creates a history record in the database even if no mails are to be sent.

JDBC Connection to SQL Server
If you are using an older version of the JDBC driver, (e.g. `msbase.jar`, `mssqlserver.jar`, `msutil.jar`), then the JDBC connection URL in the `[appdata_path]/config/factory.ini` configuration file will be different to that required by the newer `sqljdbc.jar` version.

.

The older version uses:

```
db = jdbc -class=com.microsoft.jdbc.sqlserver.SQLServerDriver
jdbc:microsoft:sqlserver://localhost:1433;selectMethod=Cursor;databaseName=scheduler
-user=scheduler -password=scheduler
```

and the newer version:

```
db = jdbc -class=com.microsoft.sqlserver.jdbc.SQLServerDriver
jdbc:sqlserver://localhost:1433;sendStringParametersAsUnicode=false;selectMethod=cur
sor;databaseName=scheduler -user=scheduler -password=scheduler
```

Please note the different classnames and use of lowercase letters in the "cursor" value.

E-mails Cannot be Sent

The sender address must be a valid address for your mail server. Problems may occur with outgoing mails if the domain part of this address is not valid for the mail server. In this case, adjust the sender address in the `log_mail_from` entry of the `[appdata_path]/config/factory.ini` configuration file.

When Does the Job Scheduler Need a Restart?

- After changes to one of the configuration files in the directory `[appdata_path]/config` have been made.
- After libraries in the directory `[install_path]/lib` have been replaced.

A restart of the Job Scheduler is not required during normal operation; this also applies for operation over longer periods of time.

The Job Scheduler is Asking for a Licence Key, Isn't this Software Open Source?

Running the Job Scheduler on Unix with the `[install_path]/bin/scheduler` binary file or on Windows with the `[install_path]/bin/scheduler.exe` file will result in the following error message:

```
SOS-1000 No licence key was found or licence key has expired. Please contact your
systems administrator or Software- und Organisations-Service GmbH, Fax +49 (30) 861
33 35, Mail info@sos-berlin.com [Scheduler].
```

To start the Job Scheduler, use the `[install_path]/bin/jobscheduler.sh` shell script (Unix) or the `[install_path]/bin/jobscheduler.cmd` command (Windows). The binary files are parameterised by the start script. One of the parameters (`-sos.ini=...`) in the `[install_path]/bin/jobscheduler_environment_variables.(cmd|sh)` script which is called by the start script addresses the `[appdata_path]/config/sos.ini` licence file which contains the licence key for the *Job Scheduler*.

The Setup Supplies a Wrong Version of *[install_path]/lib/libperl.so*

See the FAQ (http://www.sos-berlin.com/modules/cjaycontent/index.php?id=79#question_30) for the answer.

PostgreSQL requires PL/SQL

Check the languages that are available for your database by using

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

from the command line 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
```

PHP error with IIS and MySQL: A connection attempt failed because the connected party did not properly respond after a period of time

Open up %windir%\system32\drivers\etc\hosts with a text editor and comment out the line that looks like:

```
::1 localhost
```

Prefix it with a #, like so:

```
#::1 localhost
```

Save it and your PHP/MySQL connections will immediately begin working. You could also use 127.0.0.1 in your connection string instead of localhost.

Index