



Job Scheduling

JOB SCHEDULER

Installation and Configuration

July 2010

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: July 2010

Table of Contents

1 Installation	4
1.1 Installation Using the Setup Program	4
1.2 Setup Packages	6
1.3 The Sample Jobs Package	7
1.4 Setup Forms	8
1.5 Directory Structure after Installation	17
1.6 Automatic Installation	20
1.7 Database Configuration	20
1.8 Starting and stopping the Job Scheduler	21
2 Batch Installation	22
2.1 Configuration of the XML file	24
3 Multiple Installation	29
3.1 Reinstallation of the Job Scheduler	29
3.2 Installation Alongside an Already Existing Installation	29
4 The Installation of a Backup Cluster	30
5 Deinstallation	31
5.1 Removal Using the Uninstaller	31
5.2 Manual Removal on Windows	32
5.3 Manual Removal on Unix	32
6 Configuration	33
6.1 The factory.ini File	33
6.2 The scheduler.xml and scheduler.xsd Files	33
6.3 The jobscheduler_environment_variables.sh File (for Unix)	33
6.4 The custom.inc.php File	34
6.5 Configuration of the Web Server	35
7 Automatic Update Procedure	37
7.1 SchedulerUpdate Web Service	37
7.2 CheckForUpdate Job (Client)	37
7.3 Handling Multiple Updates	39
8 Operation on 64bit Systems	41
9 Troubleshooting	43
Index	45

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 20) (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 6) to 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 43) 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.sos-berlin.com/scheduler>. Windows 2000/2003/XP/Vista, Linux starting with kernel 2.4, Solaris 8/9/10, HP-UX 11.23 and 11.31 and AIX 5.3 operating systems are supported.

- **Web Server Configuration** (page 35) (optional)

The use and configuration of a web server for the *Job Scheduler* is only necessary if the *Web Interface* (page 6) package is selected during installation. This package contains an extended web interface for PHP.

1.1 Installation Using the Setup Program

The following archive files are available for download from <http://www.sos-berlin.com/scheduler>:

- **scheduler_linux.[release].tar.gz** for Linux (archive with setup program)
- **scheduler_solaris.[release].tar.gz** for Solaris (archive with setup program)
- **scheduler_win32.[release].zip** for Windows (archive with setup program)
- **scheduler_jre_win32.[release].zip** for Windows (archive with setup program incl. JRE)
- **scheduler_hpux-ia64-32.[release].tar.gz** for HP-UX Itanium (archive with setup program)
- **scheduler_hpux-pa64-32.[release].tar.gz** for HP-UX PA-RISC (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_solaris32.jar** for Solaris
- **scheduler_win32.jar** for Windows
- **scheduler_jre_win32.exe** for Windows, includes JRE 1.5

- **scheduler_hpux-ia64-32.jar** for HP-UX Itanium
- **scheduler_hpux-pa64-32.jar** for HP-UX PA-RISC
- **scheduler_aix32.jar** for AIX.

The "*jar*" setup programs require a pre-installed Java Runtime Environment, whereas the "*exe*" program includes the Java Runtime Environment (JRE 1.5).

The setup program can be started as a dialog or in batch mode (see batch installation (page 22)).

The "*jar*" programs are started using:

```
windows-shell>java -jar [download_path]\scheduler_win32.jar
unix-shell>java -jar [download_path]/scheduler_[unix_os]32.jar
```

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

The file *scheduler_jre_win32.exe* is started using a double click.

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 the installation directory.

For the rest of this documentation the installation directory will be referred to as *[install_path]*. Specification of the installation directory is followed by the Package Selection (page 6) 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 8) chapter.

After selection of the required packages, the necessary files are copied into the installation directory. 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 *[install_path]/logs* and is named *Install_V1.2_[date][time]_[series number].log*.

The Job Scheduler Web Interface 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 TCP 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 22).

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++.sl*, *[install_path]/lib/libgcc_s.sl* (HP-UX PA-RISC)
- *[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.

For Windows Users

You must have administrator privileges when you start the setup.

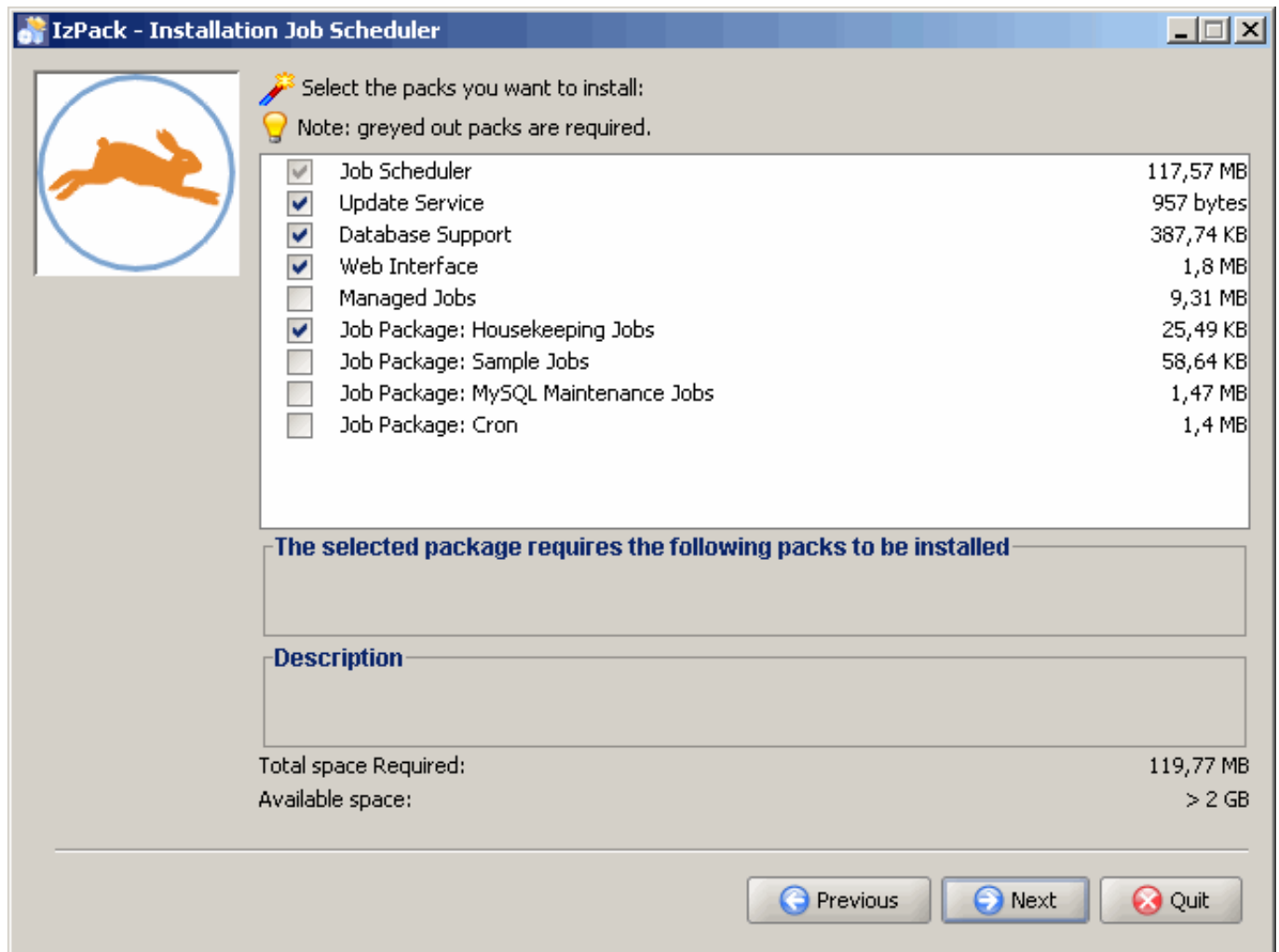
For some Windows systems you must open the console explicit with "run as administrator" even if you logged on as an user with administrator privileges.

1.2 Setup Packages

The following packages may be selected during setup:

- **Job Scheduler**
This is the basic package and must be installed.
- **Update Service**
This package inserts a job which checks every week if a new release has been made. Note that a Java Runtime Environment (JRE) must be installed before this package can be used.
- **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.
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 `./bin/import_settings.sh [language] script` or `./bin/import_settings.cmd [language]`. In both cases the `[language]` is set using either *de* for German or *en* for English (lower case).
- **Web Interface**
The Web Interface package allows the monitoring of Job Schedulers. 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.
- **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.
- **Sample Jobs**
Java, JavaScript, Perl and VBScript examples to assist in the development of Scheduler Jobs based on the API.
- **Managed Jobs**
Managed Jobs are administered in a database and are allocated to one or more Job Schedulers. 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:



1.3 The Sample Jobs Package

The example jobs are to be found after installation in the `[install_path]/samples` folder. The Job Scheduler must be made aware of these jobs manually. There is no support for this in the setup program. Instead, the `[install_path]/config/scheduler.xml` file must be edited.

It is strongly recommended to read the Job Scheduler documentation before editing this file. Errors in the `scheduler.xml` configuration file could mean that the *Job Scheduler* cannot be started. To add an example job to the `scheduler.xml` file, it is necessary to add a `<base>` element. In the following example, this is the line:

```
<base file = "../samples/config/scheduler_sample_vbscript.xml"/>
```

Example `scheduler.xml` File with Included "Sample Jobs" in VBScript:

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<spooler>
```

```
<config tcp_port = "4444"
```

```
        udp_port                = "4444"
        mail_xslt_stylesheet = "config/scheduler_mail.xsl">

<!-- included job configurations -->
<base file = "scheduler_automation_java.xml"/>
<base file = "../samples/config/scheduler_sample_vbscript.xml"/>

<!-- host name, IP address or network address of hosts, -->
<!-- that are allowed to communicate with the job scheduler -->
<security ignore_unknown_hosts = "yes">
    <allowed_host host = "localhost" level = "all"/>
</security>

<process_classes>
    <!-- max. number of processes in default process class -->
    <process_class                max_processes = "10" />
    <!-- max. number of processes running in single instances -->
    <process_class name = "single"    max_processes = "10" />
    <!-- max. number of processes running in multiple instances -->
    <process_class name = "multi"     max_processes = "10" />
</process_classes>

</config>

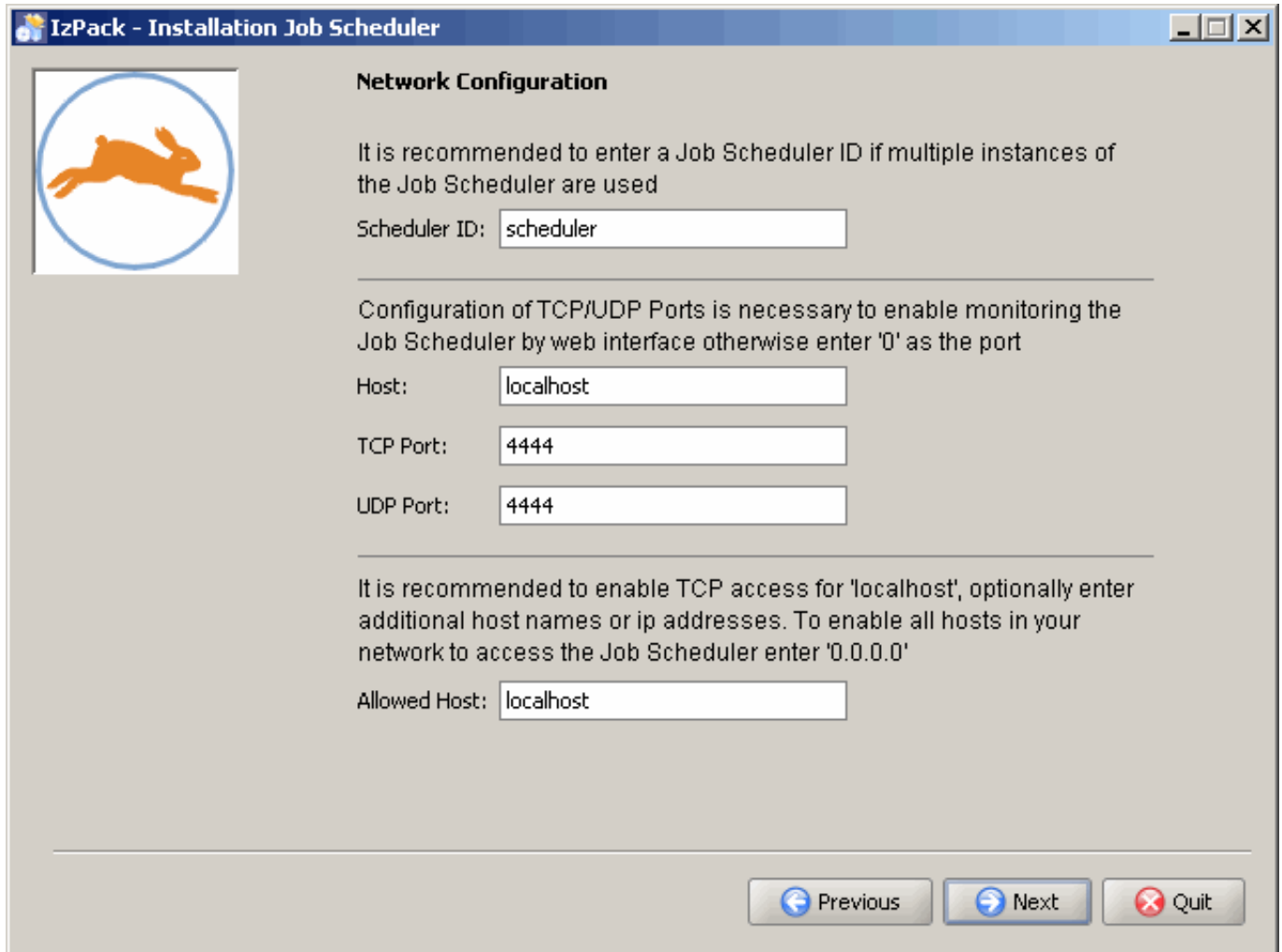
</spooler>
```

The Job Scheduler needs to be (re)started after these changes have been made. Note that an FTP Server and a Java JRE need to be installed for this JavaScript example to work.

1.4 Setup Forms

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

1.4.1 The Basic Job Scheduler Forms



IzPack - Installation Job Scheduler

Network Configuration

It is recommended to enter a Job Scheduler ID if multiple instances of the Job Scheduler are used

Scheduler ID:

Configuration of TCP/UDP Ports is necessary to enable monitoring the Job Scheduler by web interface otherwise enter '0' as the port

Host:

TCP Port:

UDP Port:

It is recommended to enable TCP access for 'localhost', 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'

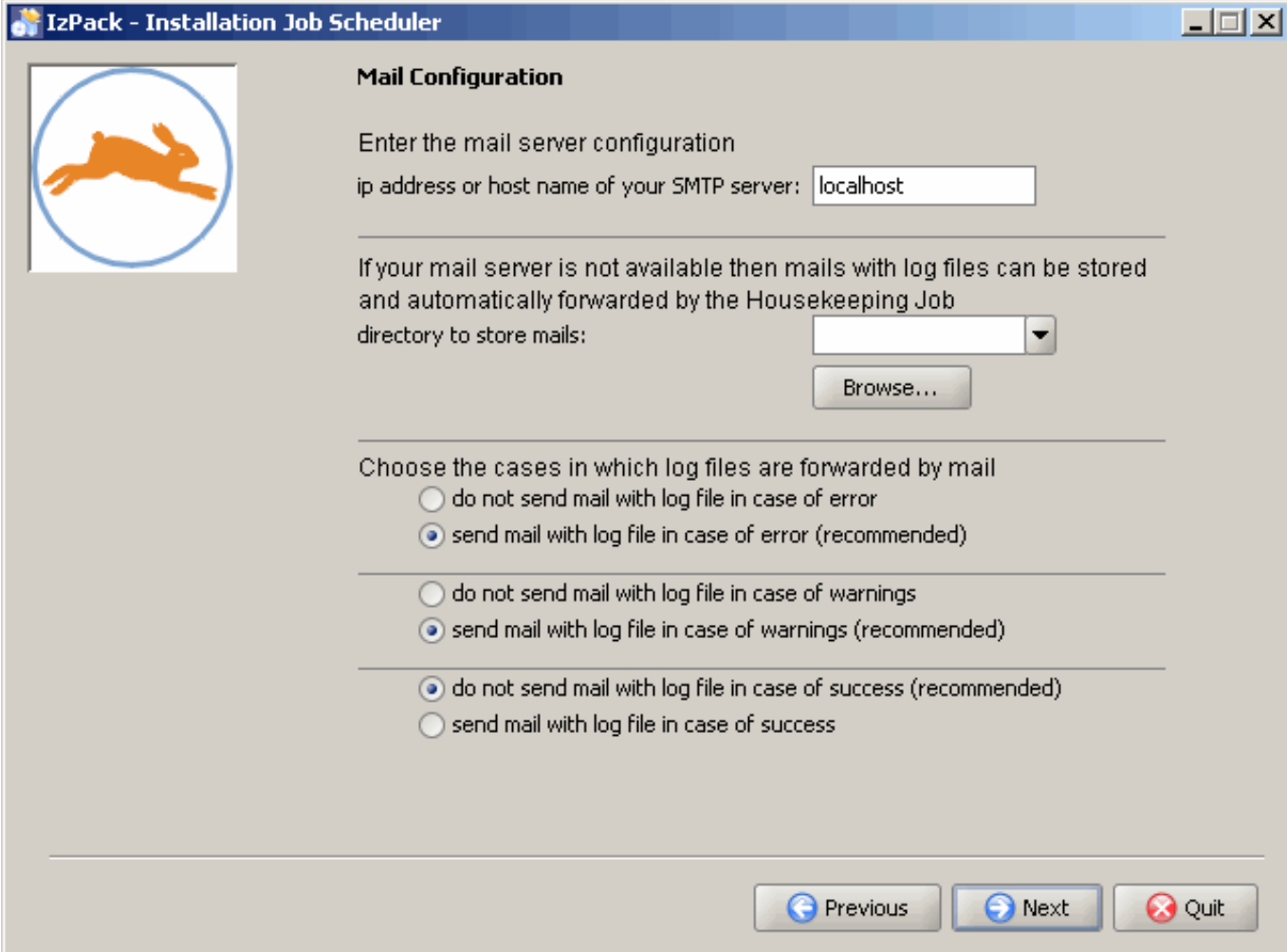
Allowed Host:

The Job Scheduler ID is entered in the *Scheduler ID* input box. Lower case letters and/or numbers are allowed here, but not special characters or symbols. 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 *TCP-Port* - is used for communication with the web interface.

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 TCP port configure the *[install_path]/web/custom/custom.inc.php* file. The Scheduler ID, TCP port, the UDP port and the Allowed Host entries are also written to the *[install_path]/config/scheduler.xml* file. Both configuration files can be changed manually (page 33) later on.



The screenshot shows the 'IzPack - Installation Job Scheduler' window. On the left is a logo of a running rabbit inside a blue circle. The main area is titled 'Mail Configuration'. It contains a text box for 'ip address or host name of your SMTP server' with 'localhost' entered. Below this is a section for 'If your mail server is not available then mails with log files can be stored and automatically forwarded by the Housekeeping Job', featuring a dropdown menu and a 'Browse...' button. Further down, there are three groups of radio buttons for selecting when to forward log files by mail: 'in case of error', 'in case of warnings', and 'in case of success'. Each group has two options: 'do not send mail' and 'send mail' (with '(recommended)' next to the latter). At the bottom right are three buttons: 'Previous', 'Next', and 'Quit'.


IzPack - Installation Job Scheduler

Mail Configuration

Enter the mail server configuration

ip address or host name of your SMTP server:

If your mail server is not available then mails with log files can be stored and automatically forwarded by the Housekeeping Job

directory to store mails: 

Choose the cases in which log files are forwarded by mail

☐ do not send mail with log file in case of error

☒ send mail with log file in case of error (recommended)

☐ do not send mail with log file in case of warnings

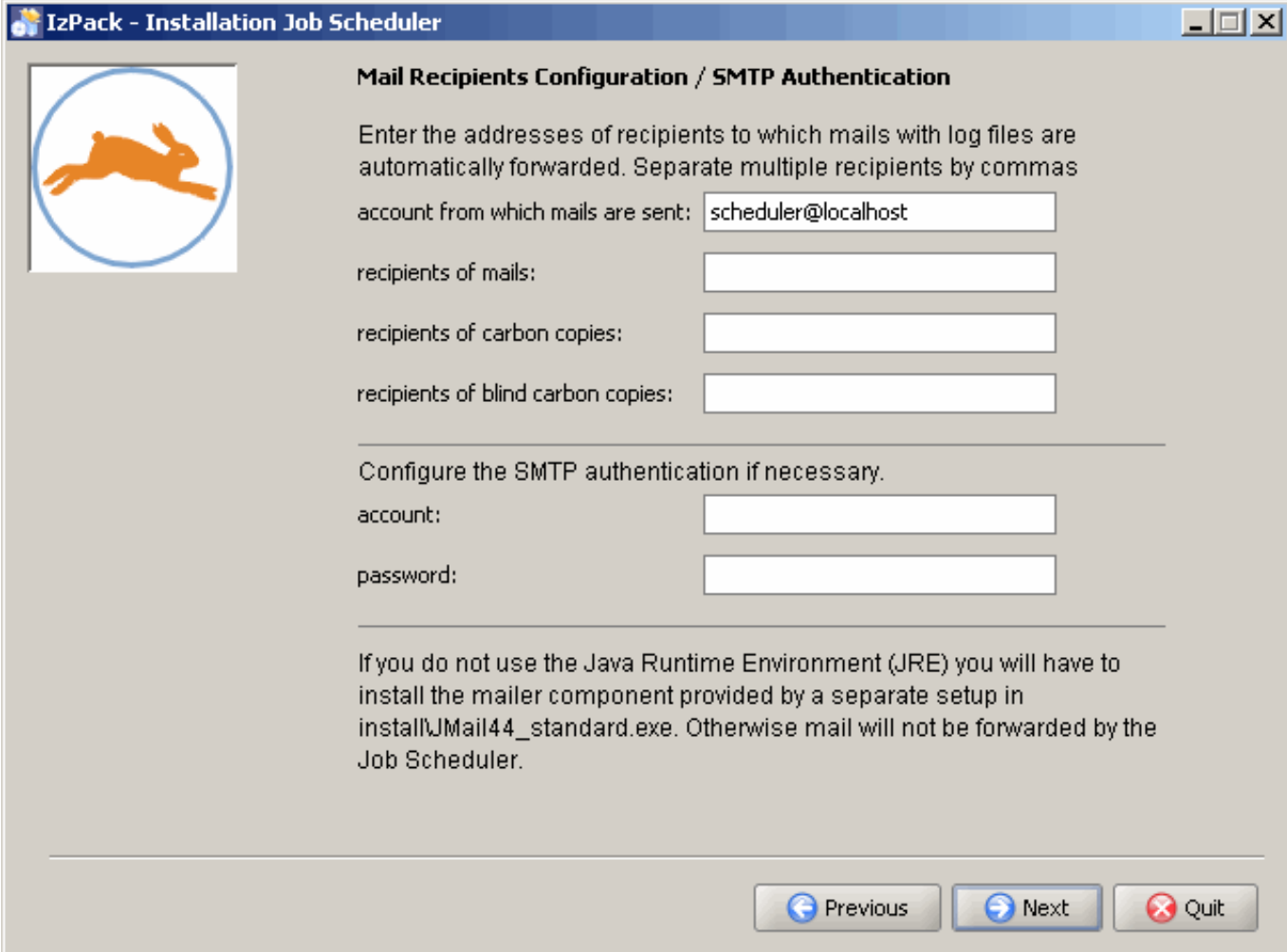
☒ send mail with log file in case of warnings (recommended)

☒ do not send mail with log file in case of success (recommended)

☐ send mail with log file in case of success

The SMTP Server is specified here along with information regarding whether the *Job Scheduler* should automatically forward job log files by e-mail.

The values entered here configure the `[install_path]/config/factory.ini` file, which can also be changed manually (page 33) at a later date.



The image shows a screenshot of the 'IzPack - Installation Job Scheduler' window. The title bar is blue with the text 'IzPack - Installation Job Scheduler' and standard window controls. On the left, there is a logo featuring a blue circle with an orange rabbit inside. The main area has a light gray background. The title 'Mail Recipients Configuration / SMTP Authentication' is in bold. Below it, a text box explains that users should enter recipient addresses for log file forwarding, separated by commas. There are three input fields: 'account from which mails are sent:' (containing 'scheduler@localhost'), 'recipients of mails:', 'recipients of carbon copies:', and 'recipients of blind carbon copies:'. A horizontal line separates this from the SMTP authentication section, which has the text 'Configure the SMTP authentication if necessary.' followed by 'account:' and 'password:' input fields. Another horizontal line follows. A paragraph at the bottom explains that if the Java Runtime Environment (JRE) is not used, the user must install the mailer component 'installJMail44_standard.exe'. At the bottom right, there are three buttons: 'Previous' (with a left arrow), 'Next' (with a right arrow), and 'Quit' (with a red X).

Mail Recipients Configuration / SMTP Authentication

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

account from which mails are sent:

recipients of mails:

recipients of carbon copies:

recipients of blind carbon copies:

Configure the SMTP authentication if necessary.

account:

password:

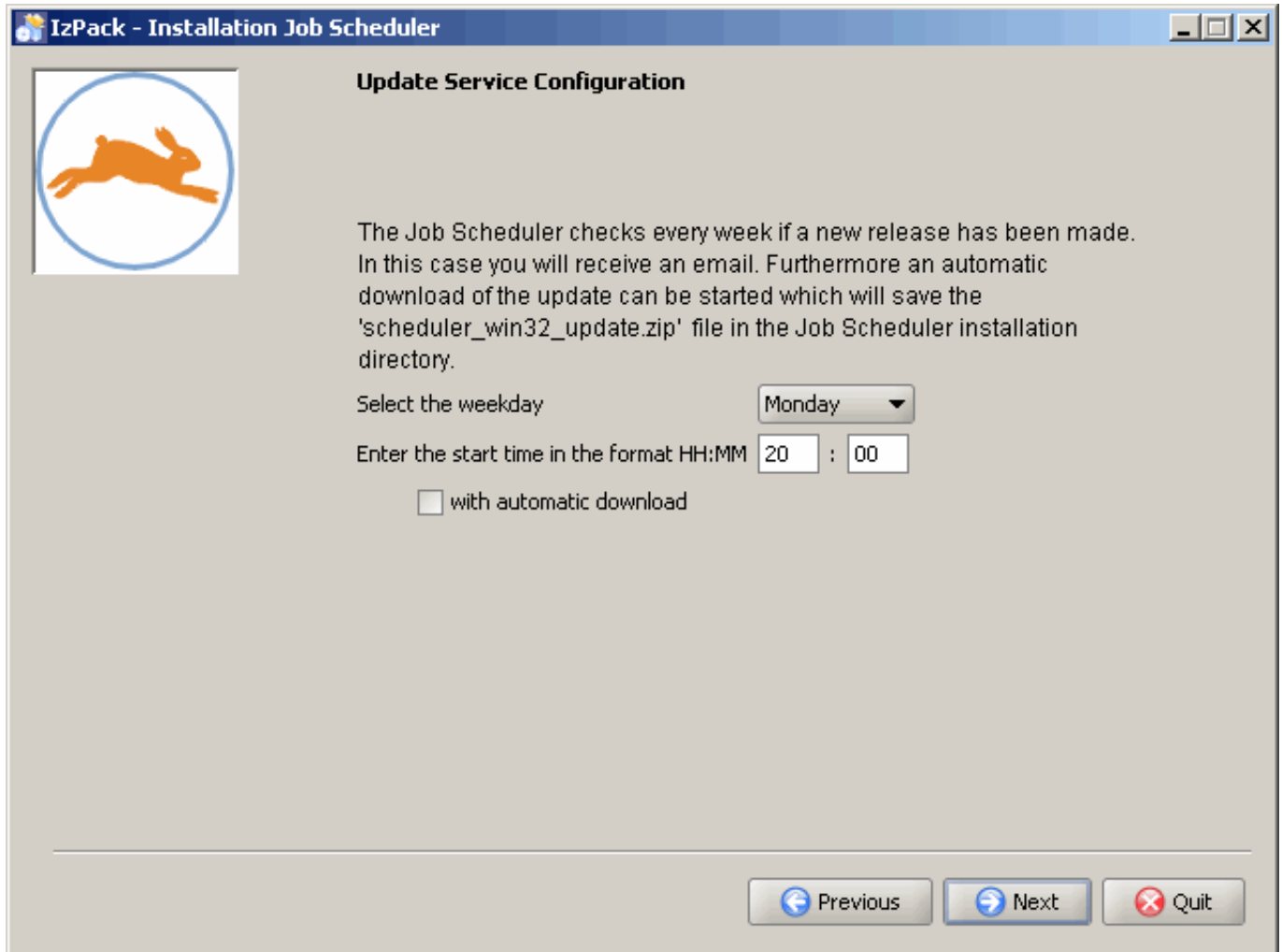
If you do not use the Java Runtime Environment (JRE) you will have to install the mailer component provided by a separate setup in `installJMail44_standard.exe`. Otherwise mail will not be forwarded by the Job Scheduler.

Job protocols are automatically forwarded by the *Job Scheduler* according to the settings made in the previous form. The mail sender, recipient and if required CC und BCC are specified in this form. Multiple addresses are to be separated by commas.

In order to use the Job Scheduler's automatic mailing on Windows systems without the Java Runtime Environment, it is necessary that the accompanying JMail is installed by starting the following file `[install_path] \installJMail44_standard.exe`.

The entries made using this form are saved in the `[install_path]/config/factory.ini` file, which can also be subsequently changed manually (page 33).

1.4.2 The Update Service Package form

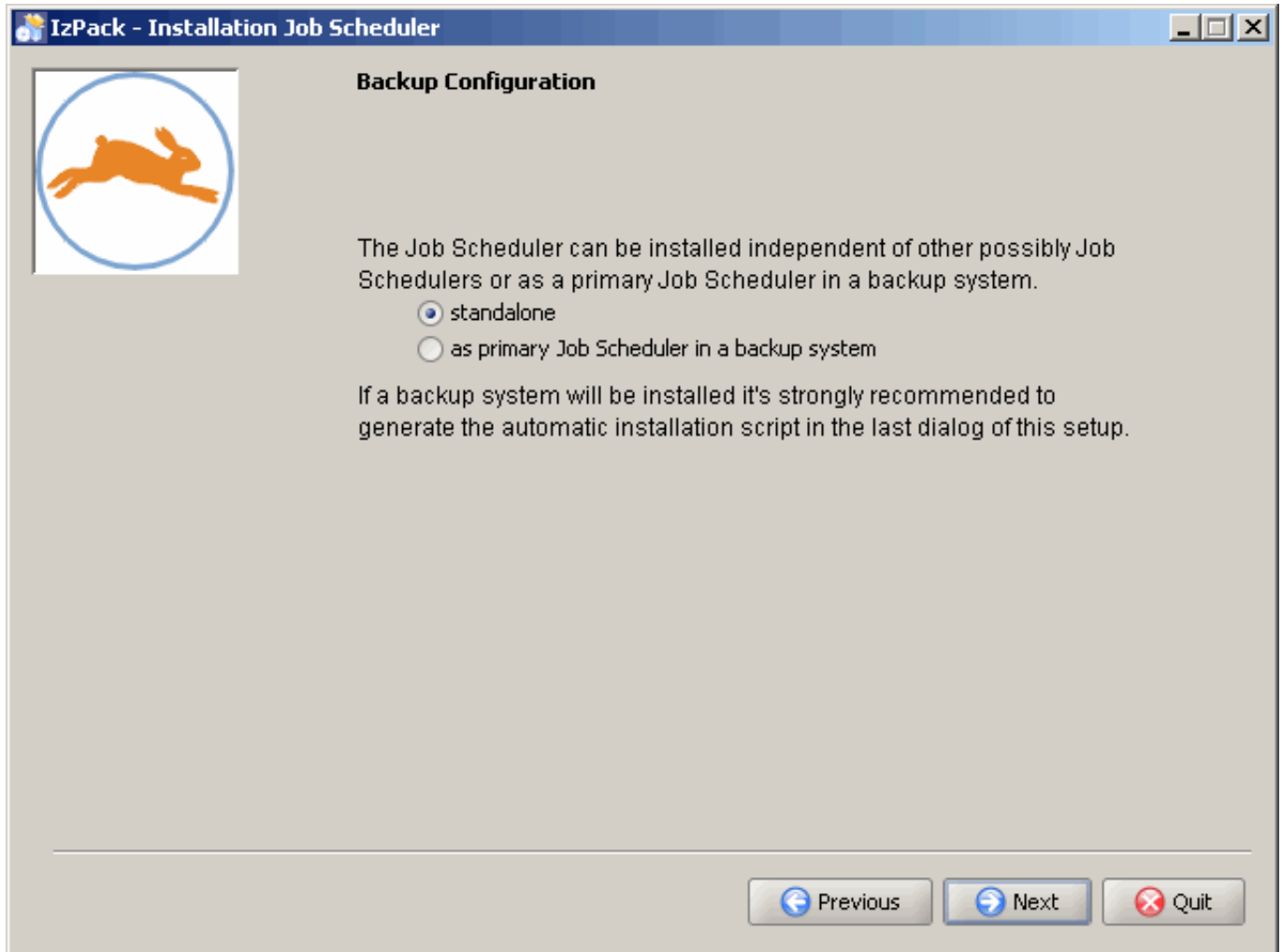


The screenshot shows a window titled "IzPack - Installation Job Scheduler". Inside, there is a section titled "Update Service Configuration". On the left, there is a logo of a running rabbit inside a circle. The main text area contains the following text: "The Job Scheduler checks every week if a new 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." Below this text, there are three configuration options: "Select the weekday" with a dropdown menu set to "Monday", "Enter the start time in the format HH:MM" with input fields showing "20" and "00", and a checkbox labeled "with automatic download" which is currently unchecked. At the bottom right, there are three buttons: "Previous", "Next", and "Quit".

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* installation path with the operating system dependent name of *scheduler_win32_update.zip*, *scheduler_linux_update.tar.gz*, *scheduler_solaris_update.tar.gz*, *scheduler_hpux-ia64-32_update.tar.gz* or *scheduler_aix_update.tar.gz*.

The *./config/scheduler_update_service.xml* file may be used for later job configuration. Further information about the Update Service can be found in the *./doc/en/scheduler_update_service.pdf* documentation.

1.4.3 The Database Support Package Forms



IzPack - Installation Job Scheduler

Backup Configuration

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

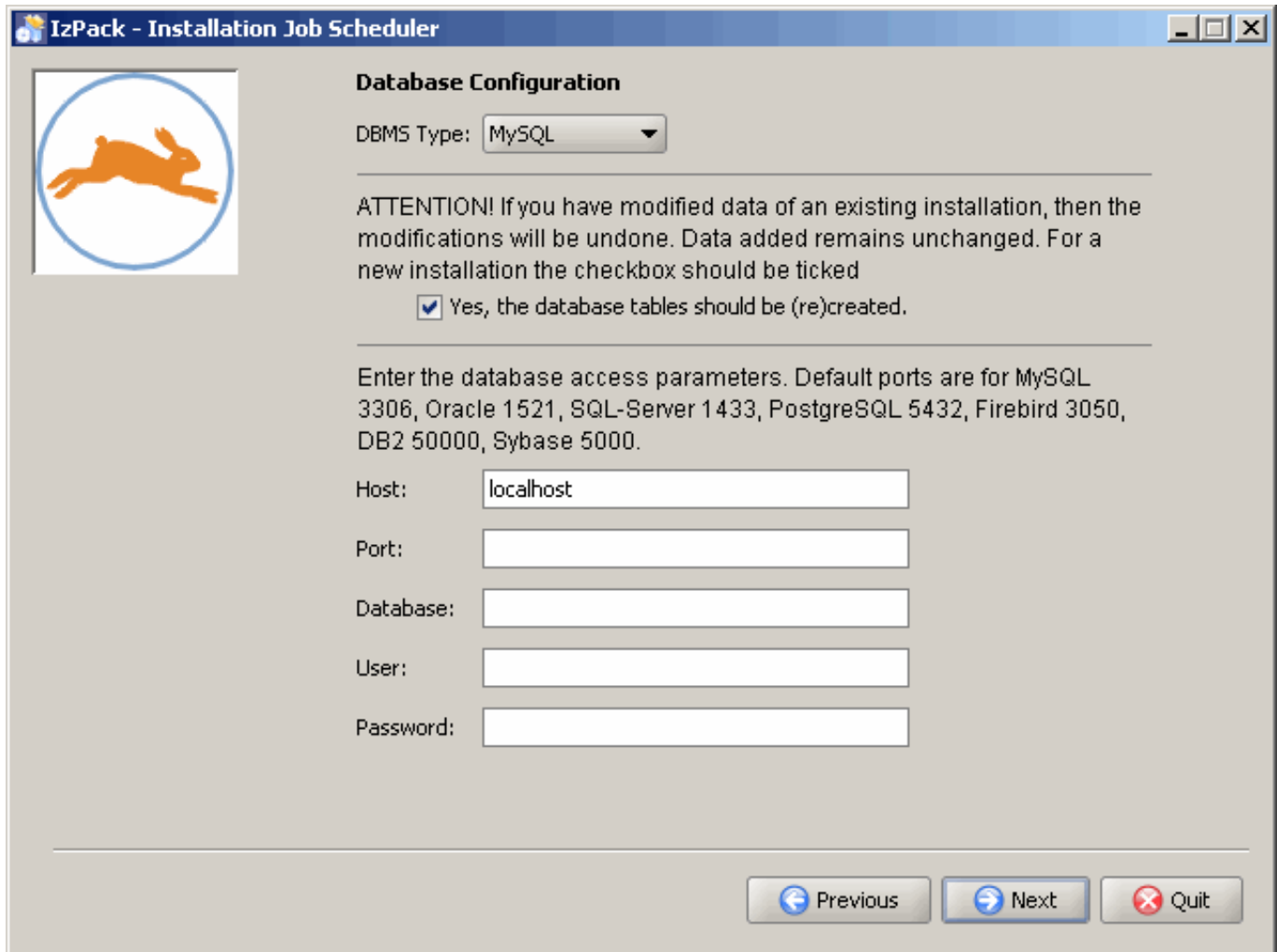
☒ standalone

☐ as primary Job Scheduler in a backup system

If a backup system will be installed it's strongly recommended to generate the automatic installation script in the last dialog of this setup.

Previous Next Quit

The radio buttons in the form shown above determine whether the *Job Scheduler* should be installed "stand-alone" or in a cluster as the primary *Job Scheduler* in a backup system. (see also Installation of a Backup System (page 30)).



The screenshot shows the 'IzPack - Installation Job Scheduler' window. On the left is a logo of a running rabbit inside a blue circle. The main area is titled 'Database Configuration'. It features a 'DBMS Type:' dropdown menu set to 'MySQL'. Below this is an 'ATTENTION!' message: 'If you have modified data of an existing installation, then the modifications will be undone. Data added remains unchanged. For a new installation the checkbox should be ticked'. A checkbox labeled 'Yes, the database tables should be (re)created.' is checked. Further down, text instructs the user to 'Enter the database access parameters. Default ports are for MySQL 3306, Oracle 1521, SQL-Server 1433, PostgreSQL 5432, Firebird 3050, DB2 50000, Sybase 5000.' Below this text are five input fields labeled 'Host:', 'Port:', 'Database:', 'User:', and 'Password:'. The 'Host:' field contains 'localhost'. At the bottom right are three buttons: 'Previous' (with a left arrow), 'Next' (with a right arrow), and 'Quit' (with a red X).

IzPack - Installation Job Scheduler

Database Configuration

DBMS Type: MySQL

ATTENTION! If you have modified data of an 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 (re)created.

Enter the database access parameters. Default ports are for MySQL 3306, Oracle 1521, SQL-Server 1433, PostgreSQL 5432, Firebird 3050, DB2 50000, Sybase 5000.

Host:

Port:

Database:

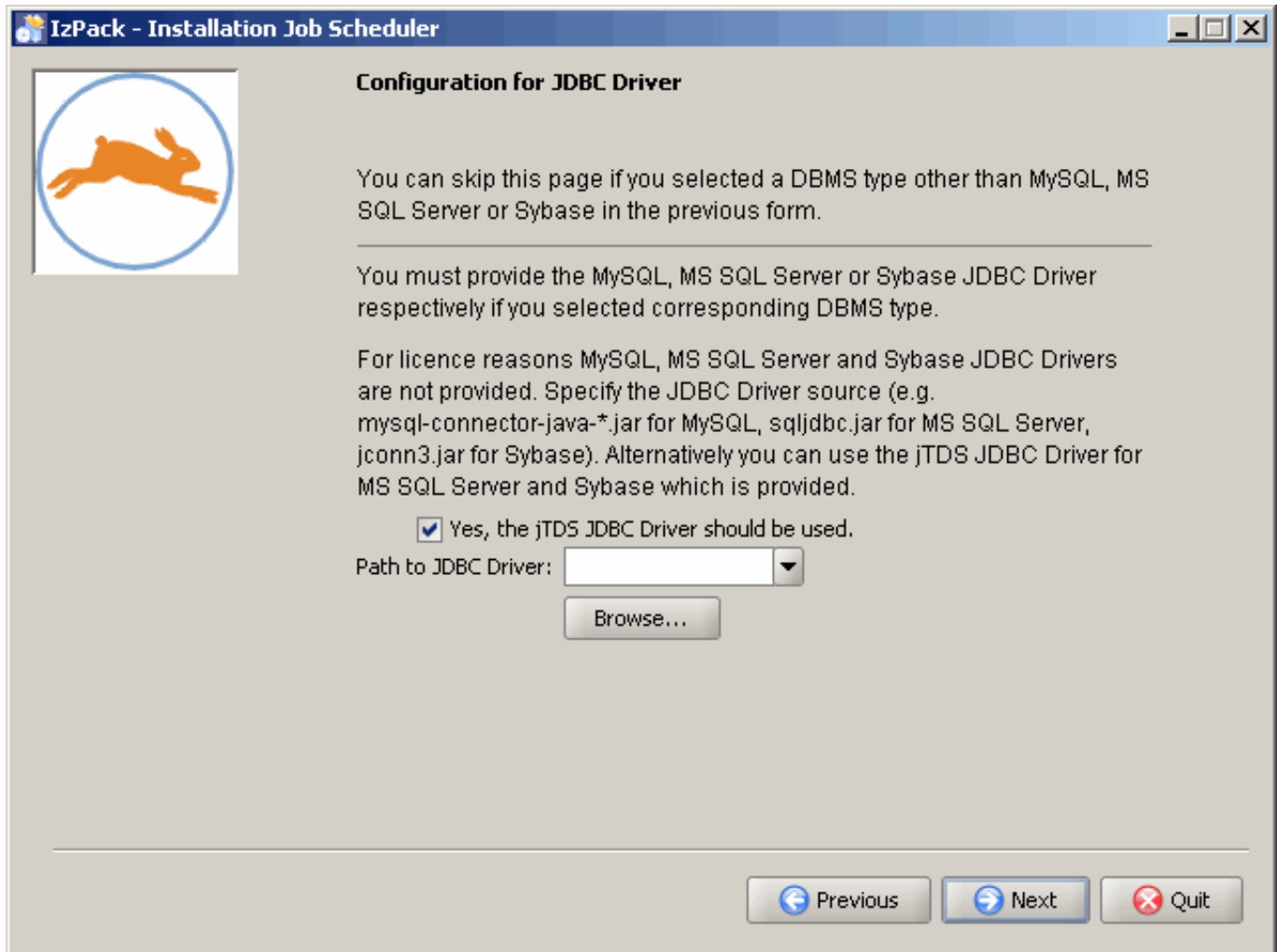
User:

Password:

Previous Next Quit

The database system is specified in the upper selection box on this form and the database connection information is specified in the input fields. 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 21).

This configuration is saved in the `[install_path]/config/factory.ini` and `[install_path]/web/custom/custom.inc.php` files. Both files can be changed manually (page 33) if required.



IzPack - Installation Job Scheduler

Configuration for JDBC Driver

You can skip this page if you selected a DBMS type other than MySQL, MS SQL Server or Sybase in the previous form.

You must provide the MySQL, MS SQL Server or Sybase JDBC Driver respectively if you selected corresponding DBMS type.

For licence reasons MySQL, MS SQL Server and Sybase 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.

☒ Yes, the jTDS JDBC Driver should be used.

Path to JDBC Driver:

This dialog form is only relevant for MySQL, Sybase and SQL Server databases and can be omitted when Oracle, PostgreSQL, Firebird or DB2 are being used. 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 this 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.

These configurations are stored in the *[install_path]/config/factory.ini* file, where they can later be changed manually (page 33).

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

1.4.4 The Housekeeping Jobs Package Form



Job Configuration

Housekeeping Jobs are provided for the following tasks: retry mail forwarding for log files in case of smtp failures, remove temporary files, restart Job Scheduler

Choose the implementation for Housekeeping Jobs

Java (recommended) ▼

☐ 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

Previous Next Quit

The Job Scheduler *Housekeeping Jobs* are implemented in Java, JavaScript, VBScript and Perl. The selection list in this form is used to select the *Housekeeping Jobs* version to be installed. Note that the scope of the *Housekeeping Jobs* depends upon the script language used.

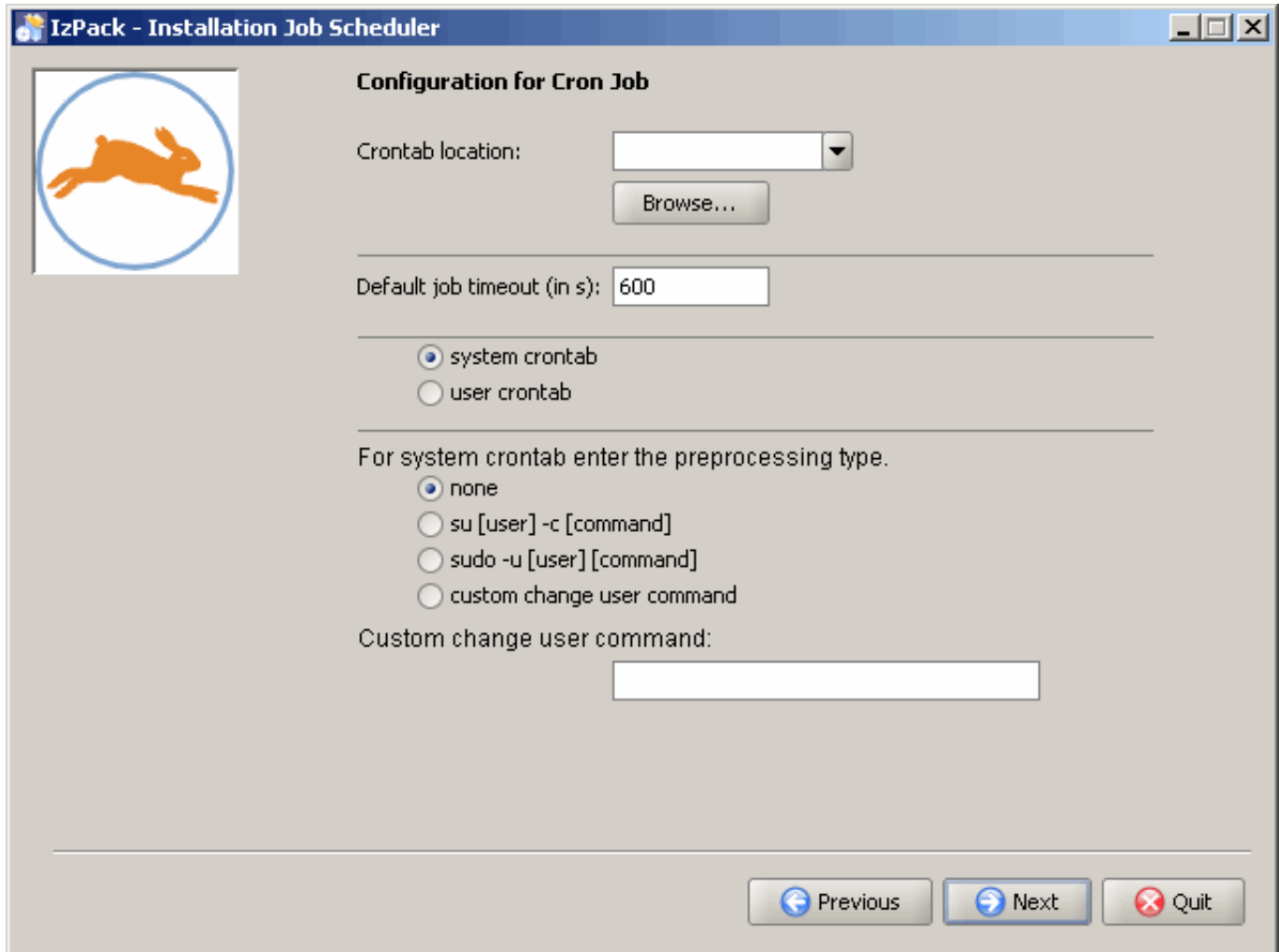
Documentation for these jobs can be found after installation in the `[install_path]/jobs` directory in XML format.

Language	Housekeeping Jobs
Java	scheduler_dequeue_mail, scheduler_restart, scheduler_rotate_log, scheduler_cleanup_history, scheduler_cleanup_files, scheduler_check_sanity
JavaScript	scheduler_dequeue_mail, scheduler_restart
VBScript	scheduler_dequeue_mail, scheduler_restart, scheduler_rotate_log, scheduler_cleanup_history
Perl	scheduler_dequeue_mail, scheduler_restart, scheduler_rotate_log

Activate the "Configure Job Scheduler as Event Handler" checkbox if Job Scheduler events are to be used and either the Job Scheduler instance being installed is the only instance that processes events or this Job Scheduler instance is a supervisor for other Job Schedulers that submit events.

The entries made in this form are saved in the `[install_path]/config/scheduler.xml`, where they can be changed manually (page 33) later on.

1.4.5 The Cron Job Packet Form



IzPack - Installation Job Scheduler

Configuration for Cron Job

Crontab location:

Default job timeout (in s):

☒ system crontab
☐ user crontab

For system crontab enter the preprocessing type.

☒ none
☐ su [user] -c [command]
☐ sudo -u [user] [command]
☐ custom change user command

Custom change user command:

1.5 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 Job Scheduler comes with its own HTTP server as a simple web interface. Note that this web interface is not the same as the more advanced PHP interface that can be selected as a package during the setup.

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

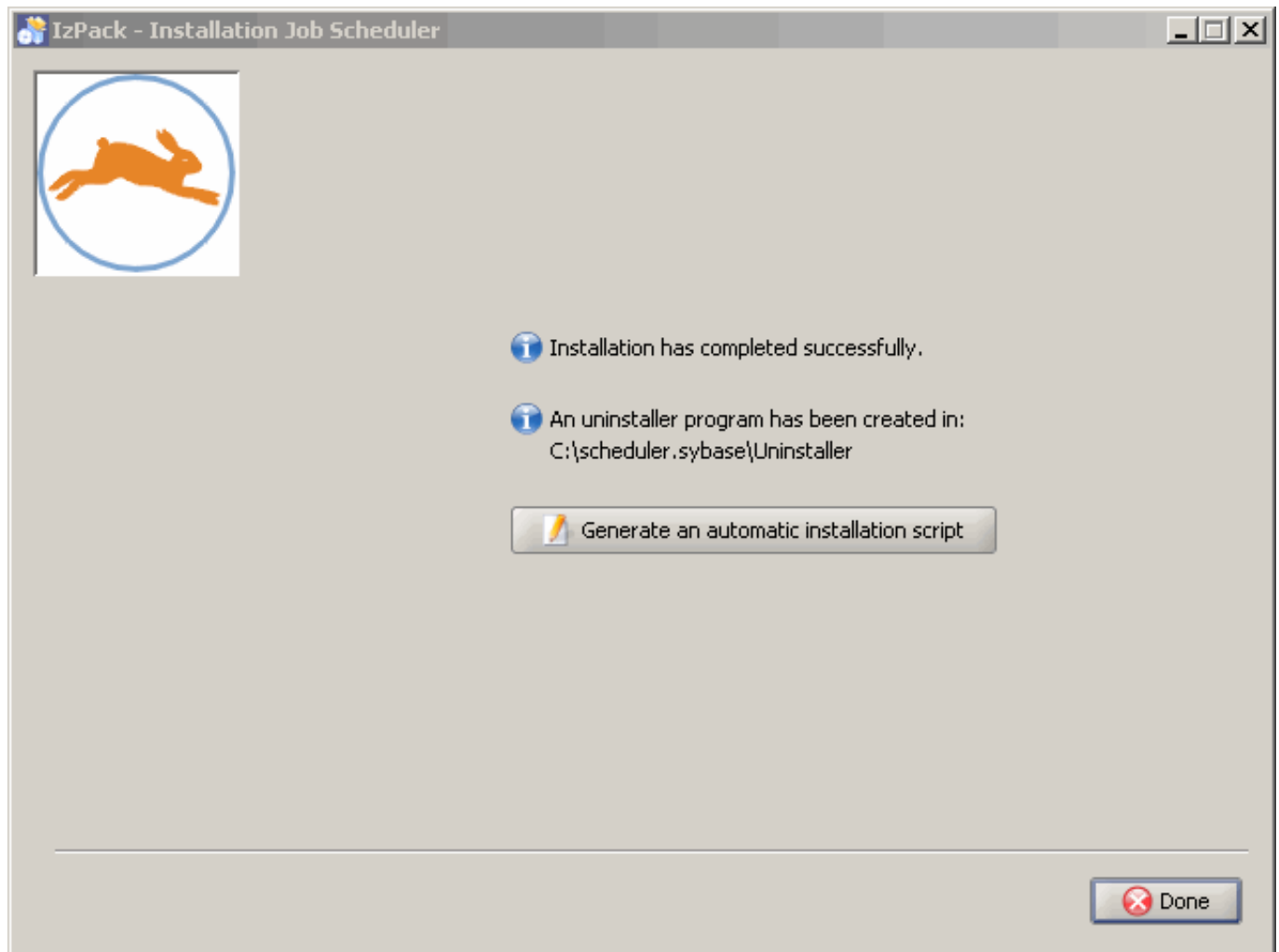
- + **bin** (Windows)
 - **hostjava.dll** Program library
 - **hostole.dll** Program library
 - **jobeditor.cmd** Start script for the Job Configuration Editor
 - **jobscheduler.cmd** Start script for the Job Scheduler
 - **jobscheduler_environment_variables.cmd** Environment variables of the Job Scheduler
 - **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
- **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
- + **config**
 - + **cache** Configuration directory in a *Workload Scheduler* (Replicate of a *Supervisor* remote directory)
 - + **events** (Housekeeping jobs/event handling)
 - + **html** The *Job Scheduler* web interface
 - + **live** Local configuration directory for the *Job Scheduler* (Hot Folder)
 - + **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 for shell scripts
 - **scheduler_update_service.xml** (Update Service)
 - **scheduler_automation_java.xml** (Housekeeping Jobs)
 - **scheduler_automation_javascript.xml** (Housekeeping Jobs)
 - **scheduler_automation_perlscript.xml** (Housekeeping Jobs)
 - **scheduler_automation_vbscript.xml** (Housekeeping Jobs)
 - **scheduler_events.xml** (Housekeeping Jobs/Event Handling)
 - **scheduler_managed.xml** (Managed Jobs)
 - **default.xslt** (Managed Jobs)
 - **mail.xslt** (Managed Jobs)
 - **scheduler_mysql.xml** (MySQL Maintenance Jobs)
 - **scheduler_mysql_javascript.xml** (MySQL Maintenance Jobs)
 - **factory_mysql.ini** (MySQL Maintenance Jobs)
 - **replication_master_settings.ini** (MySQL Maintenance Jobs)
 - **replication_slave_settings.ini** (MySQL Maintenance Jobs)
- + **db** Database objects
 - **settings_insert_en.xml** (Managed Jobs)
 - **settings_insert_de.xml** (Managed Jobs)
 - + **msaccess** MS Access
 - **scheduler.mdb**
 - **scheduler_managed.mdb** (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)
- + **jobs** Job scripts (not Java) and their documentation (HTML)
- + **lib**
 - ***.jar** Java archives (for Java jobs)
 - **scheduler.dll** for Java debugging (Windows)
 - ***.so, *.sl, *.a** libraries (Unix)
- + **logs** Depository for log files
- + **samples** (Sample Jobs)
- + **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

1.6 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.



Instructions for starting the automatic setup script can be found under Batch Installation (page 22).

1.7 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 6)* package is installed and the database connection is specified in the appropriate setup form.

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

1.7.1 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 `[install_path]/config/factory.ini`, `[install_path]/config/sos_settings.ini` and `[install_path]/web/custom/custom.inc.php` configuration files (page 33).

1.8 Starting and stopping the Job Scheduler

1.8.1 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.

1.8.2 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:

```
<AutomatedInstallation langpack="eng">
  <com.izforge.izpack.panels.HelloPanel/>
  <com.izforge.izpack.panels.InfoPanel/>
  <com.izforge.izpack.panels.LicencePanel/>
  <com.izforge.izpack.panels.TargetPanel>
    <!-- SELECT THE INSTALLATION PATH
      It must be absolute! -->
    <installpath>[:installation path of the Job Scheduler:]</installpath>
  </com.izforge.izpack.panels.TargetPanel>
  <com.izforge.izpack.panels.PacksPanel>
    <selected>
      <!-- SELECT THE PACKS YOU WANT TO INSTALL -->
      <pack index="0"/>
      <pack index="1"/>
      <pack index="2"/>
      <pack index="3"/>
      <!--pack index="4"/-->
      <pack index="5"/>
      <!--pack index="6"/-->
      <!--pack index="7"/-->
      <!--pack index="8"/-->
    </selected>
  </com.izforge.izpack.panels.PacksPanel>
  <com.izforge.izpack.panels.UserInputPanel>
    <userInput>
      <!-- Network Configuration -->
      <entry key="serviceHost" value="localhost"/>
      <entry key="serviceId" value="scheduler"/>
      <entry key="serviceAllowedHost" value="localhost"/>
      <entry key="serviceTcpPort" value="4444"/>
      <entry key="serviceUdpPort" value="4444"/>
    </userInput>
  </com.izforge.izpack.panels.UserInputPanel>
  <com.izforge.izpack.panels.UserInputPanel>
    <userInput>
      <!-- Backup Configuration -->
      <entry key="clusterOptions" value=""/>
    </userInput>
  </com.izforge.izpack.panels.UserInputPanel>
  <com.izforge.izpack.panels.UserInputPanel>
    <userInput>
      <!-- Mail Configuration -->
      <entry key="mailServer" value="localhost"/>
      <entry key="mailQueueDirectory" value=""/>
    </userInput>
  </com.izforge.izpack.panels.UserInputPanel>
</AutomatedInstallation>
```

```

        <entry key="mailOnWarning" value="yes"/>
        <entry key="mailOnSuccess" value="no"/>
        <entry key="mailOnError" value="yes"/>
    </userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel>
    <userInput>
        <!-- Mail Recipients Configuration -->
        <entry key="mailFrom" value="scheduler@localhost"/>
        <entry key="mailTo" value=""/>
        <entry key="mailBcc" value=""/>
        <entry key="mailCc" value=""/>
        <entry key="smtpAccount" value=""/>
        <entry key="smtpPass" value=""/>
    </userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel>
    <userInput>
        <!-- Update Configuration -->
        <entry key="checkForUpdateStarttime" value="20:00"/>
        <entry key="checkForUpdateStartday" value="1"/>
        <entry key="autoUpdateDownload" value="0"/>
    </userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel>
    <userInput>
        <!-- Housekeeping Job Configuration -->
        <entry key="jobAutomationLanguage"
value="scheduler_automation_java.xml"/>
        <!-- Configure Job Scheduler as event handler -->
        <entry key="jobEvents" value="off"/>
    </userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel>
    <userInput>
        <!-- Database Configuration -->
        <entry key="databaseDbms" value=""/>
        <entry key="databaseCreate" value="off"/>
        <entry key="databaseHost" value="localhost"/>
        <entry key="databasePort" value=""/>
        <entry key="databaseSchema" value="scheduler"/>
        <entry key="databaseUser" value="scheduler"/>
        <entry key="databasePassword" value="scheduler"/>
    </userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel>
    <userInput>
        <!-- Using the jTDS JDBC Driver -->
        <entry key="connectorJTDS" value="yes"/>
        <!-- Configuration for JDBC Driver -->
        <entry key="connector" value=""/>
    </userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel>
    <userInput>
        <!-- Configuration for Cron Job -->

```

```

    <entry key="cron_crontab" value="/etc/crontab"/>
    <entry key="cron_systab" value="1"/>
    <entry key="cron_timeout" value="600"/>
    <entry key="cron_changeuser" value=""/>
    <entry key="cron_changecommand" value=""/>
  </userInput>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.UserInputPanel>
  <userInput/>
</com.izforge.izpack.panels.UserInputPanel>
<com.izforge.izpack.panels.InstallPanel/>
<com.izforge.izpack.panels.ProcessPanel/>
<com.izforge.izpack.panels.FinishPanel/>
</AutomatedInstallation>

```

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

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

2.1 Configuration of the XML file

The following elements can be configured - the langpack attribute in the <AutomatedInstallation> root element; the <installpath> elements; the <pack> elements and the attribute values in the <entry> elements.

Element/Attribute	Description
langpack	The language to be used in the application is set here. Possible values are 'eng' for English and 'deu' for German.
installpath	The installation path is to be entered here.
selected	<p>The <selected> element possesses elements with the form <pack index="[integer]"/>. The corresponding set-up package will be installed for each <pack> element selected.</p> <ul style="list-style-type: none"> <pack index="0"/> is the basic <i>Job Scheduler</i> package. This package must always be specified. <pack index="1"/> is the <i>Update Service</i> package. <pack index="2"/> is the <i>Database Support</i> package. <pack index="3"/> is the <i>Web Interface</i> package. <pack index="4"/> is the <i>Managed Jobs</i> package. A pre-requisite for the selection of this package is the specification of <pack index="2"/> and <pack index="3"/>. <pack index="5"/> is the <i>Housekeeping Jobs</i> package. <pack index="6"/> is the <i>Sample Jobs</i> package. <pack index="7"/> is the <i>MySQL Maintenance Jobs</i> package. A pre-requisite for the selection of this package is the specification of <pack index="2"/>. <pack index="8"/> is the <i>Cron Job</i> package. <p>More information about the packages and the conditions of their use can be found under Setup Packages (page 6). It is extremely important that the dependencies between the packages described above are observed.</p>

Element/Attribute	Description
Network Configuration	
value of key="serviceHost"	The host name or IP address, on which the <i>Job Scheduler</i> is to be operated.
value of key="serviceId"	The specification of a Scheduler ID makes sense, when more than one <i>Job Scheduler</i> is to be operated.
value of key="serviceAllowedHost"	It is recommended that TCP access is allowed for the 'localhost', other host names or IP addresses can be specified if required. The IP address '0.0.0.0' should be used to allow all the servers in a network access.
value of key="serviceTcpPort"	The configuration of the TCP/UDP port is useful, when the <i>Job Scheduler</i> is to be monitored by way of a web interface. Should this not be the case, then '0' should be specified.
value of key="serviceUdpPort"	See serviceTcpPort
Backup Configuration	
value of key="clusterOptions"	<p>The <i>Job Scheduler</i> can be operated independently of any other <i>Job Schedulers</i>; as the primary <i>Job Scheduler</i> in a backup cluster or as a backup in a backup cluster. Note that backup clusters can only be operated in conjunction with a database. This value must remain empty if the <i>Job Scheduler</i> is to be operated without a database or independently.</p> <p>The value '-exclusive' should be specified when the <i>Job Scheduler</i> is to operate as the primary in a cluster.</p> <p>The value '-exclusive -backup -backup-precedence=[N]' should be specified if the <i>Job Scheduler</i> is to operate as a backup in a cluster, where [N] is an integer (see Installation of a Backup Cluster (page 30)).</p>
E-mail Configuration	
value of key="mailServer"	IP address or host of the SMTP server
value of key="mailQueueDirectory"	E-mails can be saved together with a job log in the event of the mail server not functioning. In this case, mails would then be forwarded at a later point in time by standard jobs. This setting specifies the directory in which mails should be temporarily saved.
value of key="mailOnWarning"	This parameter specifies whether mails containing log files should be automatically sent out in the event of a warning occurring. Possible values are 'yes' and 'no'.
value of key="mailOnSuccess"	This parameter specifies whether mails containing log files should be automatically sent out in the event of a job or order being successfully completed. Possible values are 'yes' and 'no'.
value of key="mailOnError"	This parameter specifies whether mails containing log files should be automatically sent out in the event of an error occurring. Possible values are 'yes' and 'no'.
Configuration of E-mail Recipients	
value of key="mailFrom"	The sender of log file e-mails.

<i>Element/Attribute</i>	<i>Description</i>
value of key="mailto"	The recipient(s) of log file mails. The addresses of multiple recipients should be separated by commas.
value of key="mailCc"	The carbon copy recipient(s) of log file mails. Multiple recipients should be separated by commas.
value of key="mailBcc"	The blind carbon copy recipient(s) of log file mails. The addresses of multiple recipients should be separated by commas.
value of key="smtpAccount"	This parameter sets the name of a SMTP account if necessary.
value of key="smtpPass"	This parameter sets the password of the SMTP account if necessary.
Update Service Configuration The following settings will only be evaluated, when the <code><pack index="1"/></code> element exists.	
value of key="checkForUpdateStarttime"	The <i>Job Scheduler</i> checks each week to see if a new release is available. This parameter specifies the start time of this check in HH:MM format.
value of key="checkForUpdateStartday"	The <i>Job Scheduler</i> checks each week, if a new release is available. This parameter specified the day of the week as a number (0=Sunday, 1=Monday, ..., 6=Saturday).
value of key="autoUpdateDownload"	Should a new release of the <i>Job Scheduler</i> be available, then an appropriate e-mail will be sent to the administrator. In addition, the <i>Job Scheduler</i> can start an automatic download of the release. Possible values for this parameter are 1 (automatic download) or 0 (no automatic download).
Job-Konfiguration The following settings will only be evaluated, when the <code><pack index="5"/></code> element exists.	
value of key="jobAutomationLanguage"	Standard jobs that are automatically carried out are provided for the following tasks: <ul style="list-style-type: none"> • resending mail after failure of the mail server; • deletion of temporary files and • restarting the <i>Job Scheduler</i>. Select an implementation of the standard job. The implementations available are: <ul style="list-style-type: none"> • scheduler_automation_java.xml (recommended) • scheduler_automation_javascript.xml • scheduler_automation_vbscript.xml • scheduler_automation_perl.xml
value von key="jobEvents"	Select this option when Job Scheduler Events are to be used and <ul style="list-style-type: none"> • ...this Job Scheduler instance is the only one that processes events; • ...this Job Scheduler instance is a Supervisor for other Job Schedulers that starts events. Possible values are 'on' and 'off'.
Database Configuration The following settings will only be evaluated when the <code><pack index="2"/></code> element exists.	

Element/Attribute	Description
value of key= "databaseDbms"	Specifies the database system. Possible values are: <ul style="list-style-type: none"> 'mysql' (for MySQL), 'oracle' (for Oracle), 'mssql' (for MS SQL Server), 'pgsql' (for PostgreSQL), 'fbsql' (for Firebird), 'db2' (for DB2) and 'sybase' (for Sybase).
value of key= "databaseCreate"	Specifies whether the database tables should be (new) created. Possible values are: 'on' (tables are created) and 'off' (tables are not created).
value of key= "databaseHost"	The host name or the IP address of the database computer.
value of key= "databasePort"	The port for the database connection. The default ports used by different databases are: <ul style="list-style-type: none"> MySQL 3306, Oracle 1521, SQL-Server 1433, PostgreSQL 5432, Firebird 3050, DB2 50000 and Sybase 5000.
value of key= "databaseSchema"	Specifies the name of the database.
value of key= "databaseUser"	The user name for database access.
value of key= "databasePassword"	The user password for database access.
<p>Configuration of the JDBC Driver</p> <p>The following settings will only be evaluated, when the <code><pack index="2"/></code> element exists.</p> <p>If MySQL, MS SQL Server or Sybase have been selected as DBMS type, the appropriate JDBC driver must be obtained and installed separately. JDBC drivers for these databases are not delivered with the Job Scheduler installation package for licensing reasons.</p>	
value von key= "connectorJTDS"	Should a MS SQL Server or Sybase DBMS have been selected, it is possible to use the jTDS-JDBC driver delivered with the Job Scheduler. Possible values for this key are <code>yes</code> (for use of the jTDS driver) or <code>no</code> .
value von key= "connector"	This parameter need only be set when either a MySQL, MS SQL Server or a Sybase database has been selected, in which case, the appropriate JDBC driver must be installed. (The JDBC drivers for MySQL or MS SQL Server cannot be included in the <i>Job Scheduler</i> installation program because of licensing reasons.) The path to the JDBC driver is specified in this parameter (e.g. <code>mysql-connector-java-*.jar</code> for MySQL, <code>sqljdbc.jar</code> for MS SQL Server, <code>jconn3.jar</code> for Sybase). This parameter should be left blank when the jTDS-JDBC driver described above is used for MS SQL Server or Sybase databases as described above. See also <code>connectorJTDS</code> .
<p>Configuration for Cron Job</p> <p>This input panel is ONLY FOR UNIX AVAILABLE. The values will be ignored under Windows.</p>	

<i>Element/Attribute</i>	<i>Description</i>
value of key= "cron_crontab"	Enter the crontab path.
value of key= "cron_systab"	Select system crontab (1) or user crontab (0).
value of key= "cron_timeout"	Enter the default job timeout (in s). The value must greater than 0.
value of key= "cron_changeuser"	For system crontabs enter the preprocessing type <ul style="list-style-type: none">• (empty) = none• su = su [user] -c [command]• sudo = sudo -u [user] [command]• (empty) = custom change user command
value of key= "cron_changecommand"	Enter the custom change user command if 'cron_changeuser' is empty.

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.

Not all the information entered in the setup forms during a first installation is (fully) recovered when reinstalling the Job Scheduler.

With the exception of the Scheduler ID, which is entered in the first form, the three forms of the basic *Job Scheduler* package (page 9) installation can be missed out. Should changes be required in the *network* and / or *e-mail configuration*, then these must be carried out manually (page 33).

The *database configuration* (page 13) form must be completely filled out, even when no changes are made to the database connection.

When the table creation check box is marked, then the setup program executes a script that creates database tables only if they do not already exist. However, all table entries made during the original installation are set back to their original values. Data added after the first installation remains unchanged.

When the database system is to remain MySQL or MS SQL respectively, then it is not necessary to respecify the JDBC driver in the *Database Configuration* form.

Changes in the *Housekeeping Jobs* (page 16) form will be accepted.

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 9) 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 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 *main_scheduler* is defined in the *[install_path]/config/scheduler.xml* (page 33) file. The host and port of the *main_scheduler* should then be entered in the *[install_path]/web/custom/custom.inc.php* (page 34) file manually.

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 22). 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 Schedulers* 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' attribute 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*.

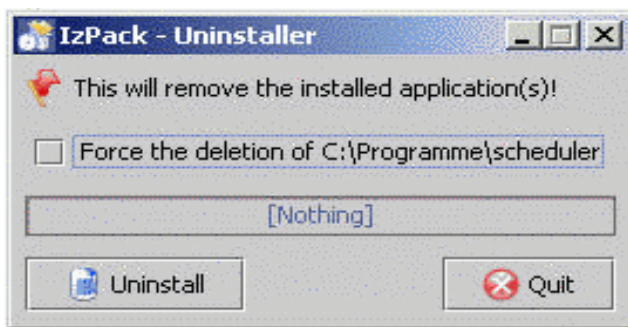
5 Deinstallation

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.



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. Note that [install_path] is used to denote the path to the Job Scheduler installation directory.

- **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]`

5.3 Manual Removal on Unix

To manually remove the Job Scheduler, a shell should be opened and then the following steps carried out. Note that the path to the *Job Scheduler* installation directory is denoted using [install_path].

- **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]`

6 Configuration

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

- **factory.ini**
- **scheduler.xml**
- **custom.inc.php** (configures the web interface)
- **jobscheduler_environment_variables.sh** (for Unix)

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 *[install_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 *[install_path]/config* directory. The *Job Scheduler* host and port information are to be found in the *scheduler.xml* file, along with details of jobs, job run times, job chains and process classes. 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 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 jobscheduler_environment_variables.sh File (for Unix)

The *jobscheduler_environment_variables.sh* file is relevant only 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 AIS 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
```

6.4 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 6) 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' ); }
```

6.5 Configuration of the Web Server

Selection of the *Web Interface* (page 6) 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 directories *[install_path]/web* and *[install_path]/logs* are available, where the virtual directory for *[install_path]/logs* must point within the *[install_path]/web* virtual directory. 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/logs/ [install_path]/logs/
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.

Installation Requirements

This job requires a Java Runtime Environment (JRE).

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 ./config/.version</release>
  <os>Operating system from Java System request</os>
  <os_install>Operating system from ./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 *./config/.version*
- the e-mail address to which the notification is to be sent is read from the *./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<
/ws: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 `./config.version` file. After the update has been downloaded, the directory in which it is saved is also noted in this file. The `./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 PA-RISC, 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 could use the JRE 1.5 that is included in the Job Scheduler installation or install a separate 32bit JRE. 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\jre1.5.0_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 *.\config\sos.ini* configuration file:
[java]
vm = C:/Program Files (x86)/Java/jre1.5.0_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=$HOME/jdk1.5.0_06/jre/bin:$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:
`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 *./bin/jobscheduler_environment_variables.sh* script:

```
LD_LIBRARY_PATH=$SCHEDULER_HOME/lib:/usr/local/lib:
$HOME/jdk1.5.0_06/jre/lib/i386:$HOME/jdk1.5.0_06/jre/lib/i386/client:$LD
_LIBRARY_PATH
```

- Job Scheduler setup on 64bit platforms

Ensure that you are using a 32bit Java JRE when executing the setup. Depending from the target platform there are important differences when installing the 32bit Job Scheduler on a 64bit platform:

- Windows 64bit

As for every 32bit program, you have to install the Job Scheduler and Java JRE in the *C:\Program Files (x86)* folder (or language specific folder name), as this folder is reserved by the operating system for 32bit programs. Installations in different folders will not work.

Should this location not be suitable for your Job Scheduler configuration and log files, then you can modify the path for these files in the *./bin/jobscheduler_environment_variables.cmd* script by changing the command line startup options (*-config*, *-log-dir*, *-log* etc.). To implement such changes, the Windows service has to be re-installed. This could be done using the following commands (where the installation directory is *C:\Program Files (x86)\scheduler*):

```
C:\>cd "C:\Program Files (x86)\scheduler\bin"
C:\Program Files (x86)\scheduler\bin>jobscheduler.cmd remove
C:\Program Files (x86)\scheduler\bin>jobscheduler.cmd install
```

- Unix 64bit

Installation of the 32bit Job Scheduler on this platform is identical to installation on 32-bit Unix. However, changes to the startup script might be required by the Java JRE location (see above).

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 ./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=cursor;
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 `./config/factory.ini` configuration file.

When Does the Job Scheduler Need a Restart?

- After changes to one of the configuration files in the directory `./config` have been made.
- After libraries in the directory `./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 `./bin/scheduler` binary file or on Windows with the `./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 `./bin/jobscheduler.sh` shell script (Unix) or the `./bin/jobscheduler.cmd` command (Windows). The binary files are parameterised by the start script. One of the parameters (`-sos.ini=...`) in the `./bin/jobscheduler_environment_variables.(cmd|sh)` script which is called by the start script addresses the `./config/sos.ini` licence file which contains a free licence key for the GPL version of the *Job Scheduler*.

Note that there is no functional difference between the GPL and commercially supported versions of the Job Scheduler. This is a technical licence key that helps us identify commercial customers in case of support incidents.

The Setup Supplies a Wrong Version of `./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

C

Configure database connection 18
Configure Database driver 15
Configure database driver 43
Configure jobs 16, 33, 33
Create database objects 18
custom.inc.php 9, 14

D

Database 6, 14, 20, 32
Database configuration 20, 6, 20, 43
database deinstallation 31
Database removal 32
Datenbank überwachen 6

E

Edit job configuration 17, 18, 33, 33

F

factory.ini 10, 11, 14, 15, 18, 33

I

Import database settings 18
Import databese settings 18

J

Job configuration 18, 29
Jobs konfigurieren 7
jobscheduler.sh 33

R

Remove service 32

S

scheduler.xml 9
Start the Job Scheduler 17, 18
Starting the Job Scheduler 21, 21
Stopping the Job Scheduler 21, 21, 32, 32

U

Uninstaller 19, 31

V

Validate job configuration 18, 33

W

Web interface configuration 21, 21, 29, 33, 34

Web Server 4, 19, 31, 32, 35

Web-Oberfläche konfigurieren 19