Job Scheduling

JOB SCHEDULER

Installation and Configuration

August 2007

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: Januar 2007

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 19) (optional)

  The Job Scheduler can be used without a database, which, however, means that job protocols and job histories will be stored to disk. The advantage of a database is that the Job Scheduler *Web Interface (page 5)* can then be used. Amongst the advantages of this interface is that it allows the retrieval of old job protocols. 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 and DB2 database systems are supported by the Job Scheduler.

  Because of licensing restrictions when used with MySQL or MS SQL databases, a JDBC driver appropriate to the database version used must be provided by the end users themselves. The corresponding drivers for Oracle, PostgreSQL, Firebird and DB2 are delivered with the Job Scheduler setup.

  Note that the notices about the choice of an appropriate JDBC driver and the configuration of a MySQL database server in ANSI mode are important and should be read. These notices can be found in the Troubleshooting (page 39) chapter.

- **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 2003/2003/XP, Linux starting with kernel 2.4, and Solaris 8/9/10 operating systems are supported.

- **Web Server Configuration** (page 32) (optional)

  The use and configuration of a web server is necessary if the *Web Interface (page 5)* package is selected during installation. The package contains an extended web interface for PHP.

  Independently from your web server the Job Scheduler has a built-in web server with a simple HTML interface to start and monitor jobs which is less functional but ready to run.

1.1 Installation Using the Setup Program

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

- **scheduler_linux.tar.gz** for Linux (archive with setup program)
- **scheduler_solaris.tar.gz** for Solaris (archive with setup program)
- **scheduler_win32.zip** for Windows (archive with setup program)
- **scheduler_jre_win32.zip** for Windows (archive with setup program incl. JRE)

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

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 the batch (see batch installation (page 21)).

The "*jar*" programs are started using:

```
windows-shell>java -jar [download_path]\scheduler_win32.jar
linux-shell>java -jar [download_path]/scheduler_linux32.jar
solaris-shell>java -jar [download_path]/scheduler_solaris32.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, 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 5) 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 with the following URL entered 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 Linux/Solaris 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 21).
The following libraries are required by the Job Scheduler:

- [install_path]/lib/libstdc++.so.6.0.3 (Linux)
- [install_path]/lib/libstdc++.so.5.0.4 (Solaris)
- [install_path]/lib/libgcc_s.so.1

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
The "*jar*" program can be started with a double click when "jar" files are linked to:

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

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 to use this package.

- **Database Support**
  This package allows the job history and job protocols to be saved in a database. MySQL, Oracle, SQL Server, PostgreSQL, Firebird 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 set the language for the PHP interface.

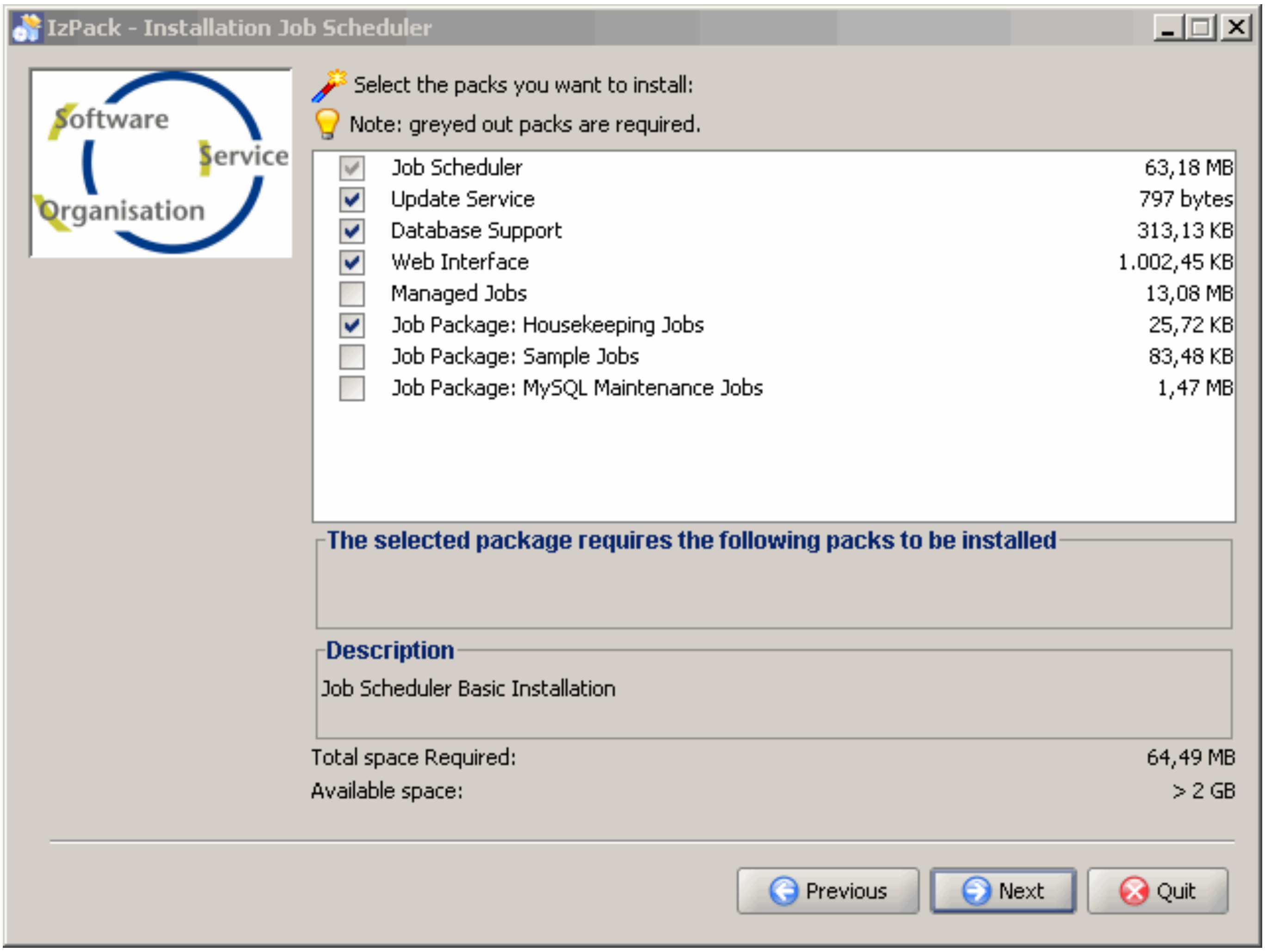
- **Housekeeping Jobs**
  Housekeeping jobs are automatically carried out by the Job Scheduler, for example, to send again temporarily stored protocol mails after failure of a mail server, to delete temporary files or to restart the Job Scheduler automatically.

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

Package selection is made using the following dialog.



Select the packs you want to install:
Note: greyed out packs are required.

☑	Job Scheduler	63,18 MB
☑	Update Service	797 bytes
☑	Database Support	313,13 KB
☑	Web Interface	1.002,45 KB
□	Managed Jobs	13,08 MB
☑	Job Package: Housekeeping Jobs	25,72 KB
□	Job Package: Sample Jobs	83,48 KB
□	Job Package: MySQL Maintenance Jobs	1,47 MB

The selected package requires the following packs to be installed

Description
Job Scheduler Basic Installation

Total space Required: 64,49 MB
Available space: > 2 GB

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 means 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 spooler_id           = "scheduler"
```

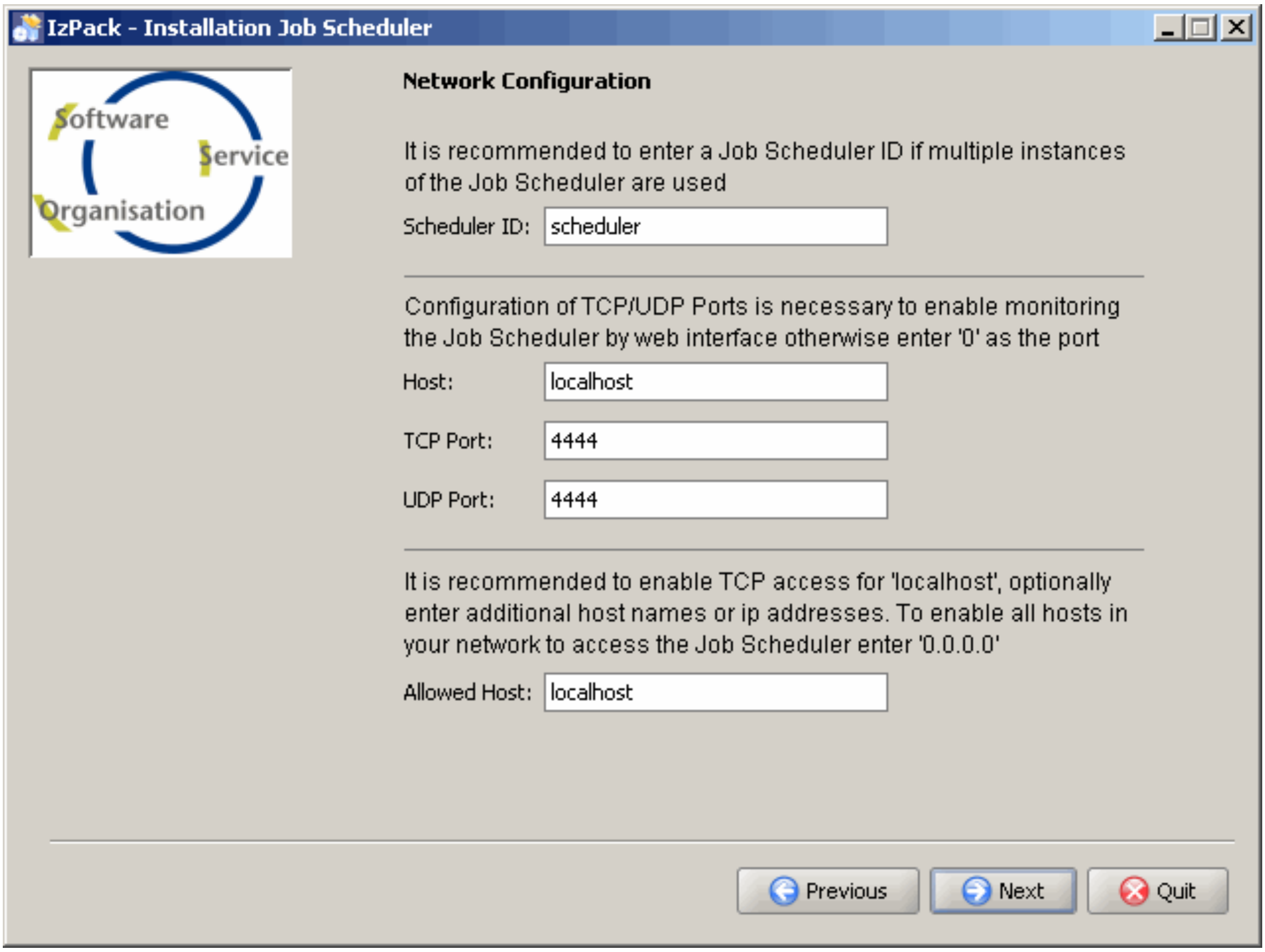
```
            tcp_port             = "4444"
            udp_port             = "4444"
            mail_xslt_stylesheet = "config/scheduler_mail.xsl">

      <!-- 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



Network Configuration

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

Scheduler ID: scheduler

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

Host: localhost

TCP Port: 4444

UDP Port: 4444

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'

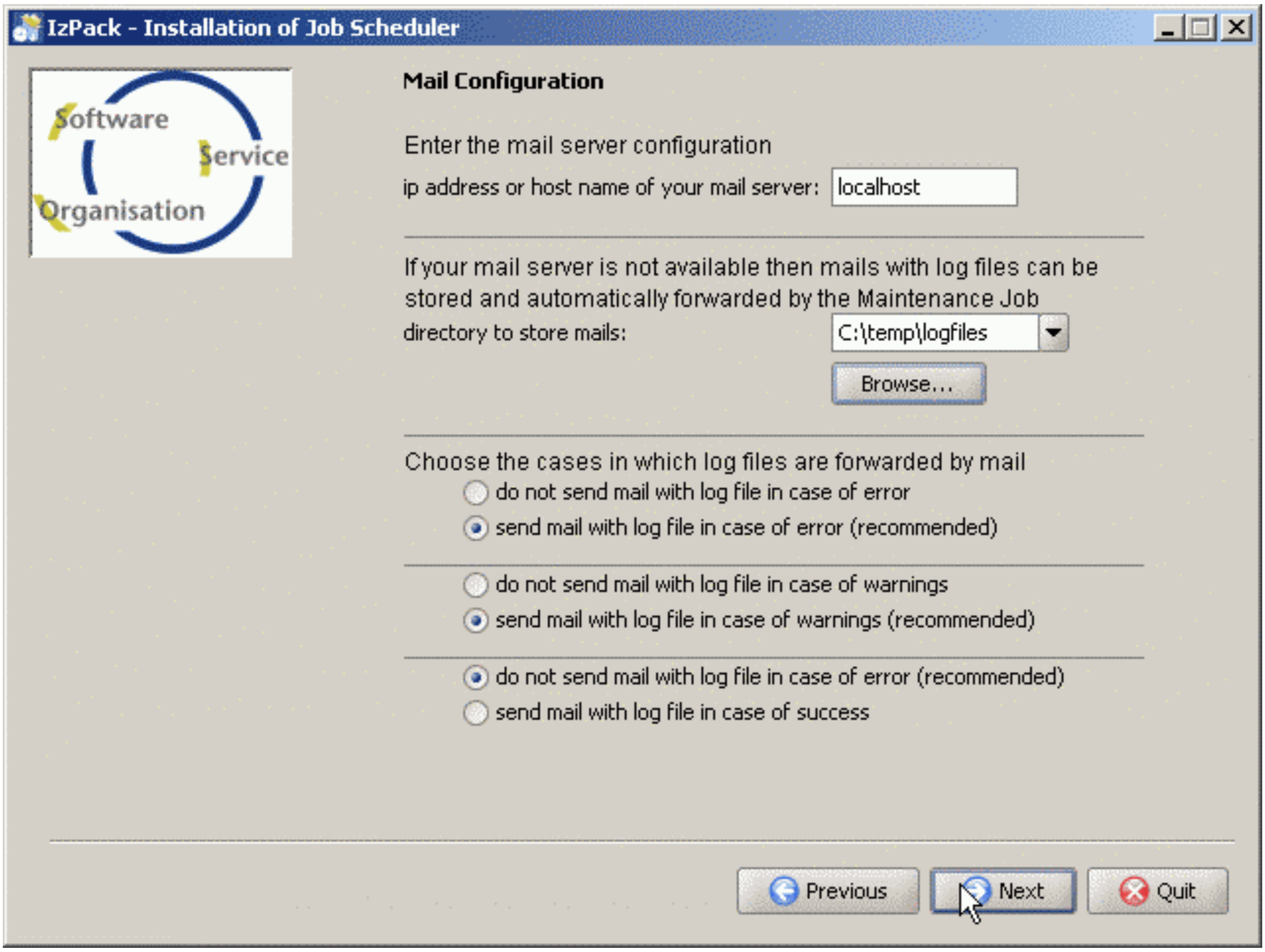
Allowed Host: localhost

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 30) later on.



IzPack - Installation of Job Scheduler

Mail Configuration

Enter the mail server configuration

ip address or host name of your mail server: localhost

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

directory to store mails: C:\temp\logfiles

Choose the cases in which log files are forwarded by mail

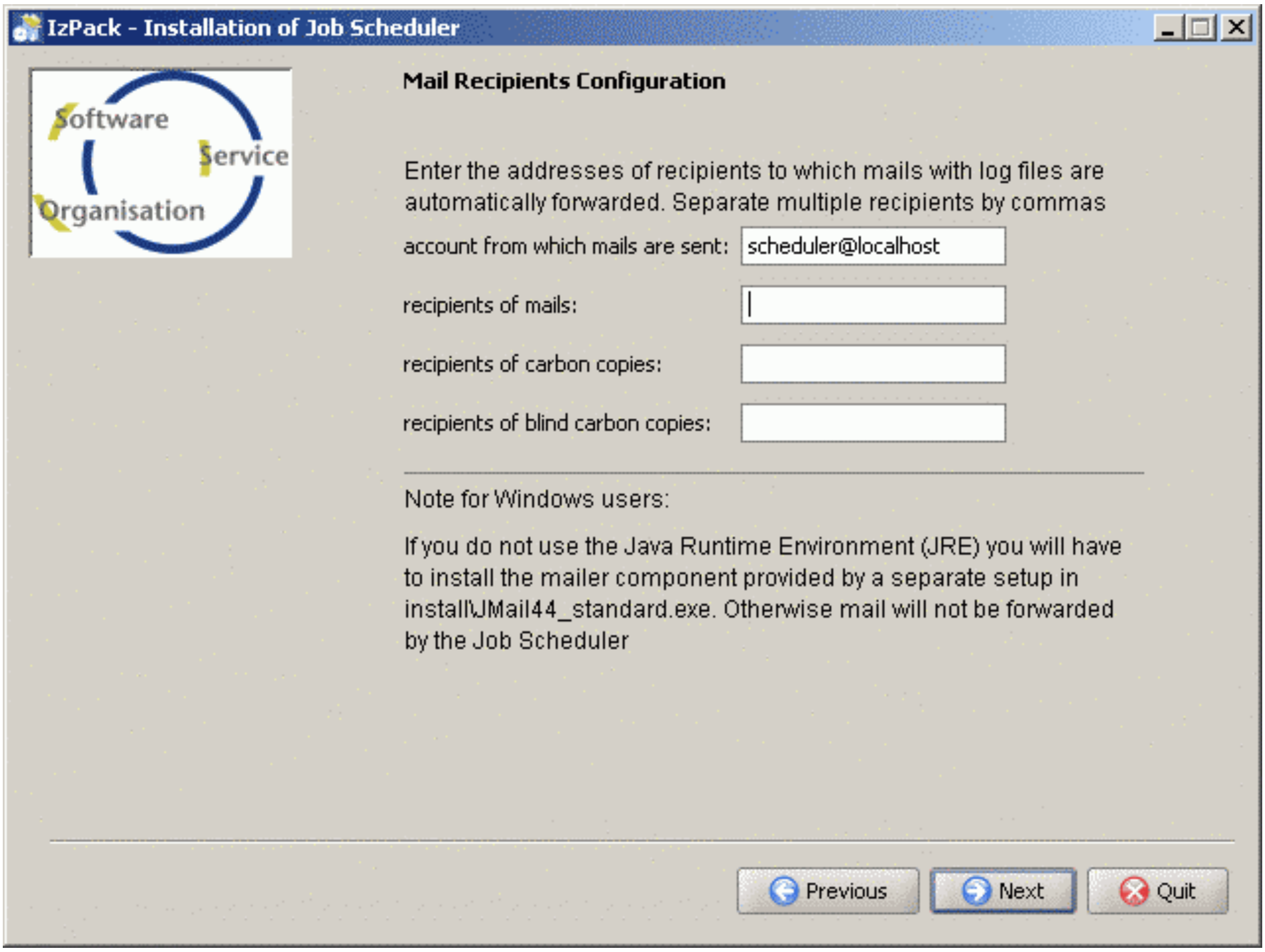
○ do not send mail with log file in case of error
● send mail with log file in case of error (recommended)

○ 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 error (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 30) at a later date.



IzPack - Installation of Job Scheduler

Mail Recipients Configuration

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@localhost

recipients of mails:

recipients of carbon copies:

recipients of blind carbon copies:

Note for Windows users:

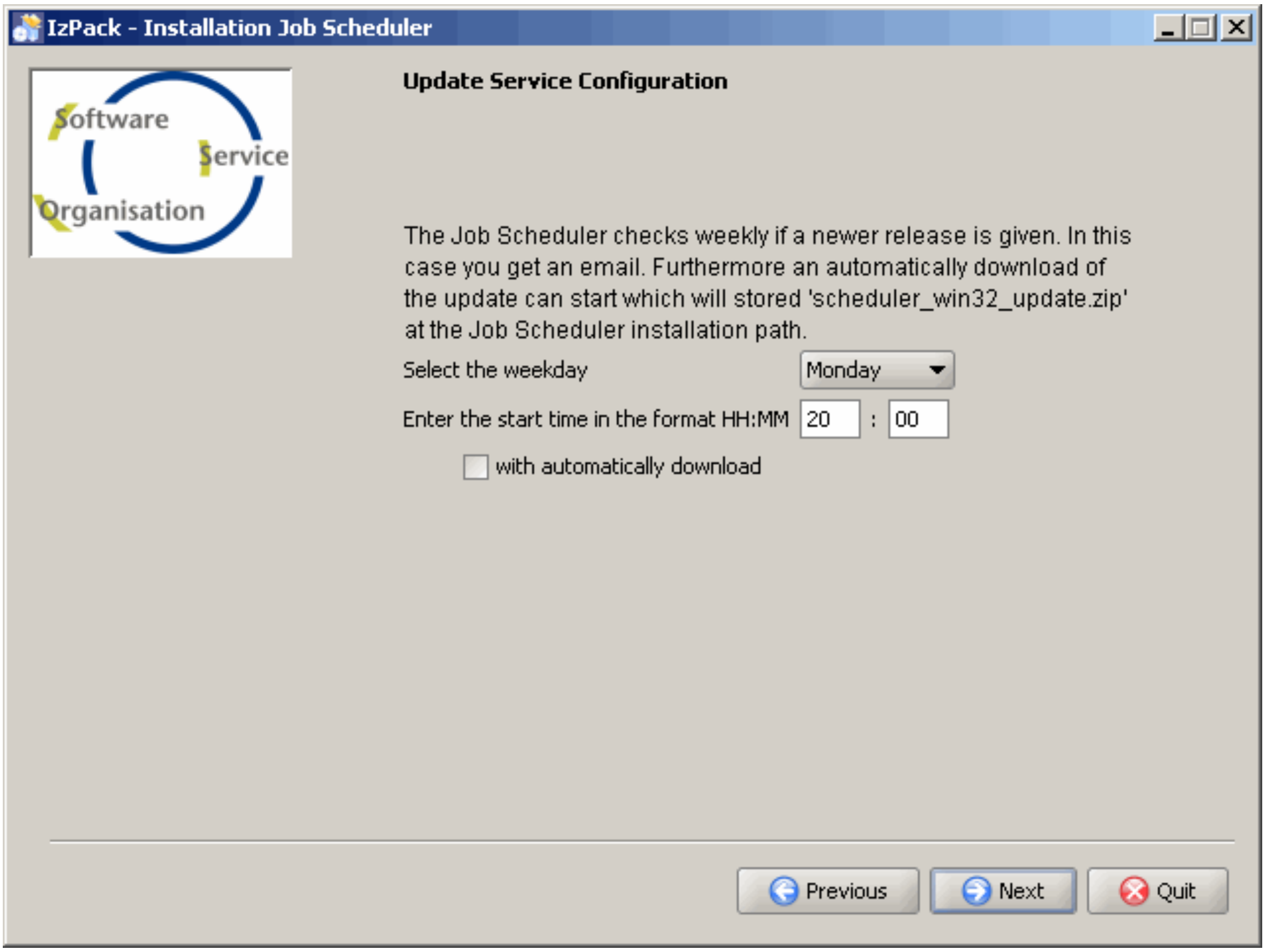
If you do not use the Java Runtime Environment (JRE) you will have to install the mailer component provided by a separate setup in install\JMail44_standard.exe. Otherwise mail will not be forwarded by the Job Scheduler

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]\install\JMail44_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 30).

1.4.2 The Update Service Package form


IzPack - Installation Job Scheduler
Software Service Organisation
Previous
Next
Quit

Update Service Configuration

The Job Scheduler checks weekly if a newer release is given. In this case you get an email. Furthermore an automatically download of the update can start which will stored 'scheduler_win32_update.zip' at the Job Scheduler installation path.

Select the weekday: Monday

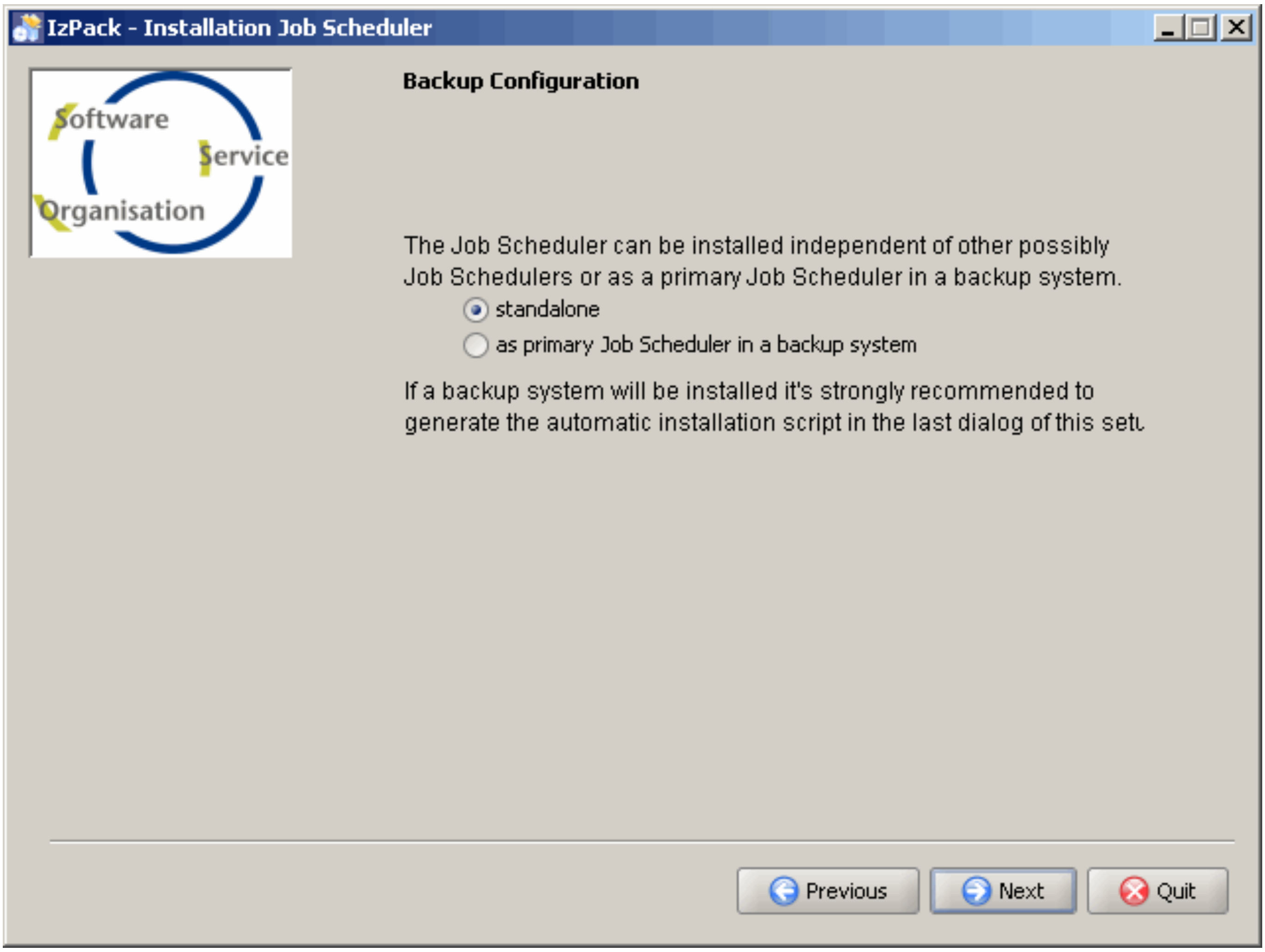
Enter the start time in the format HH:MM: 20 : 00

□ with automatically 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* installation path with the operating system dependent name of *scheduler_win32_update.zip*, *scheduler_linux_update.tar.gz* or *scheduler_solaris_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



Backup Configuration

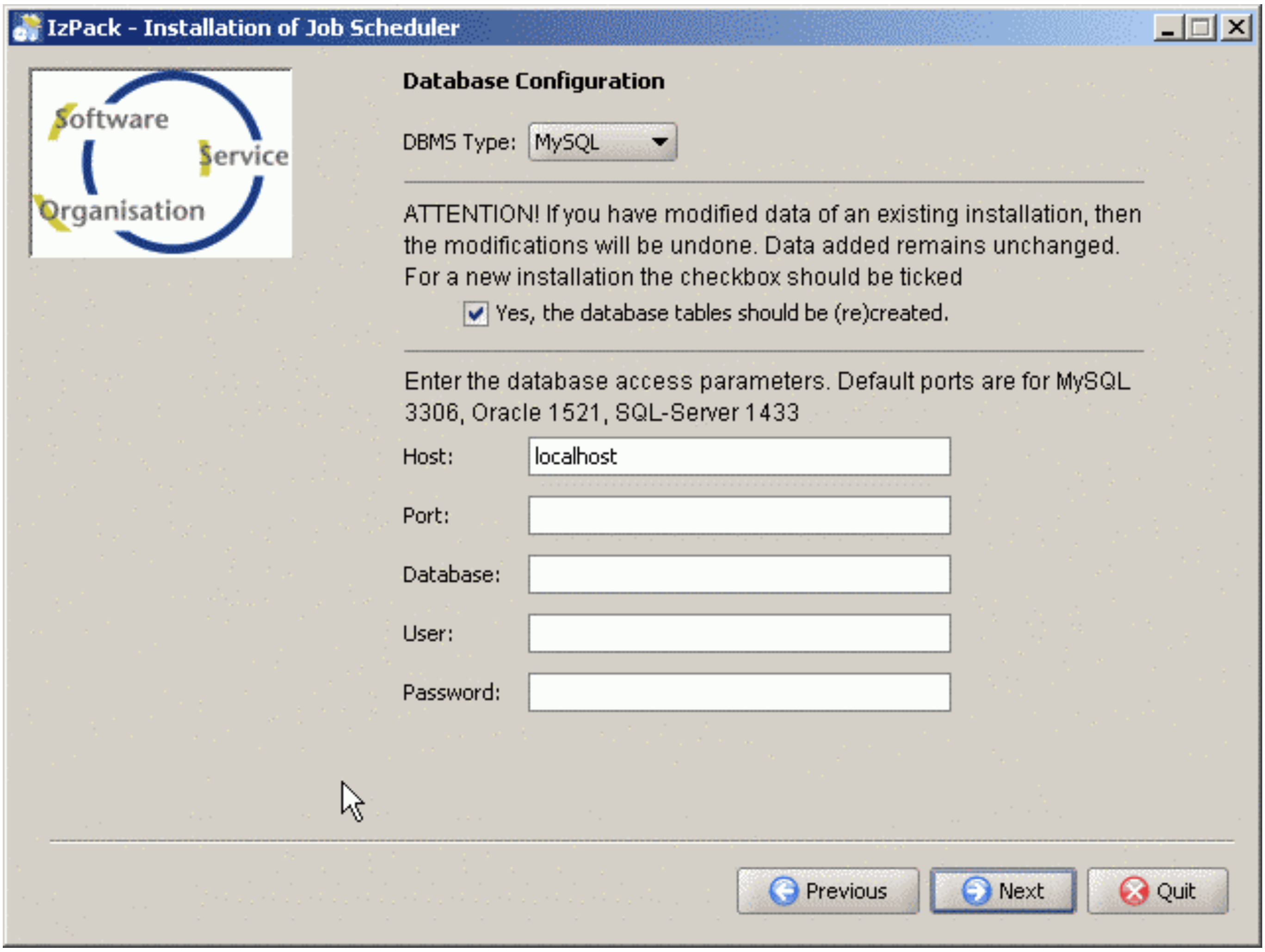
The Job Scheduler can be installed independent of other possibly Job Schedulers or as a primary Job Scheduler in a backup system.

● 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 setu

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 eines Backup Systems (page 27)).



IzPack - Installation of 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

Host: localhost

Port:

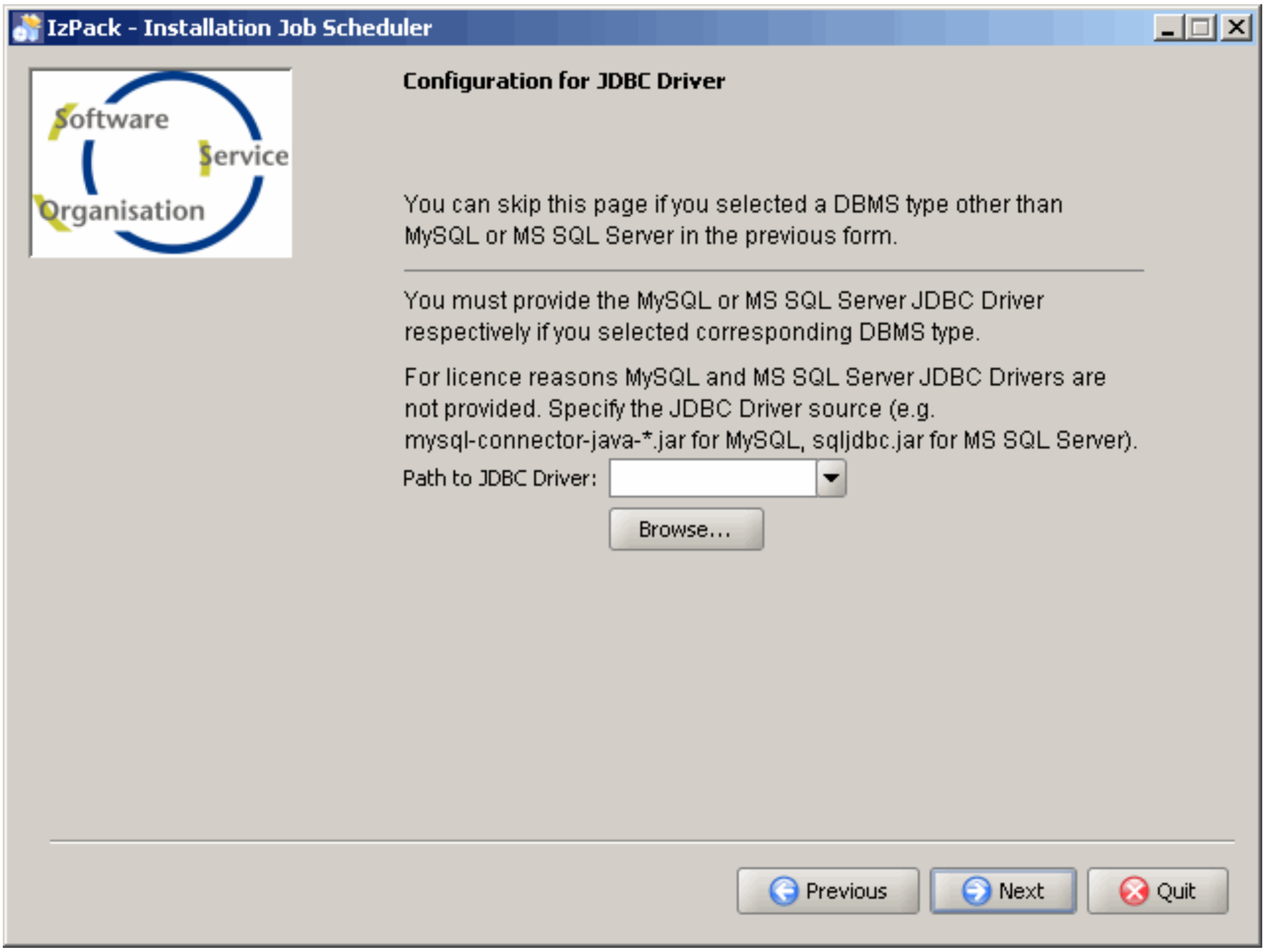
Database:

User:

Password:

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 20).

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 30) if required.



Configuration for JDBC Driver

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

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

For licence 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).

Path to JDBC Driver:

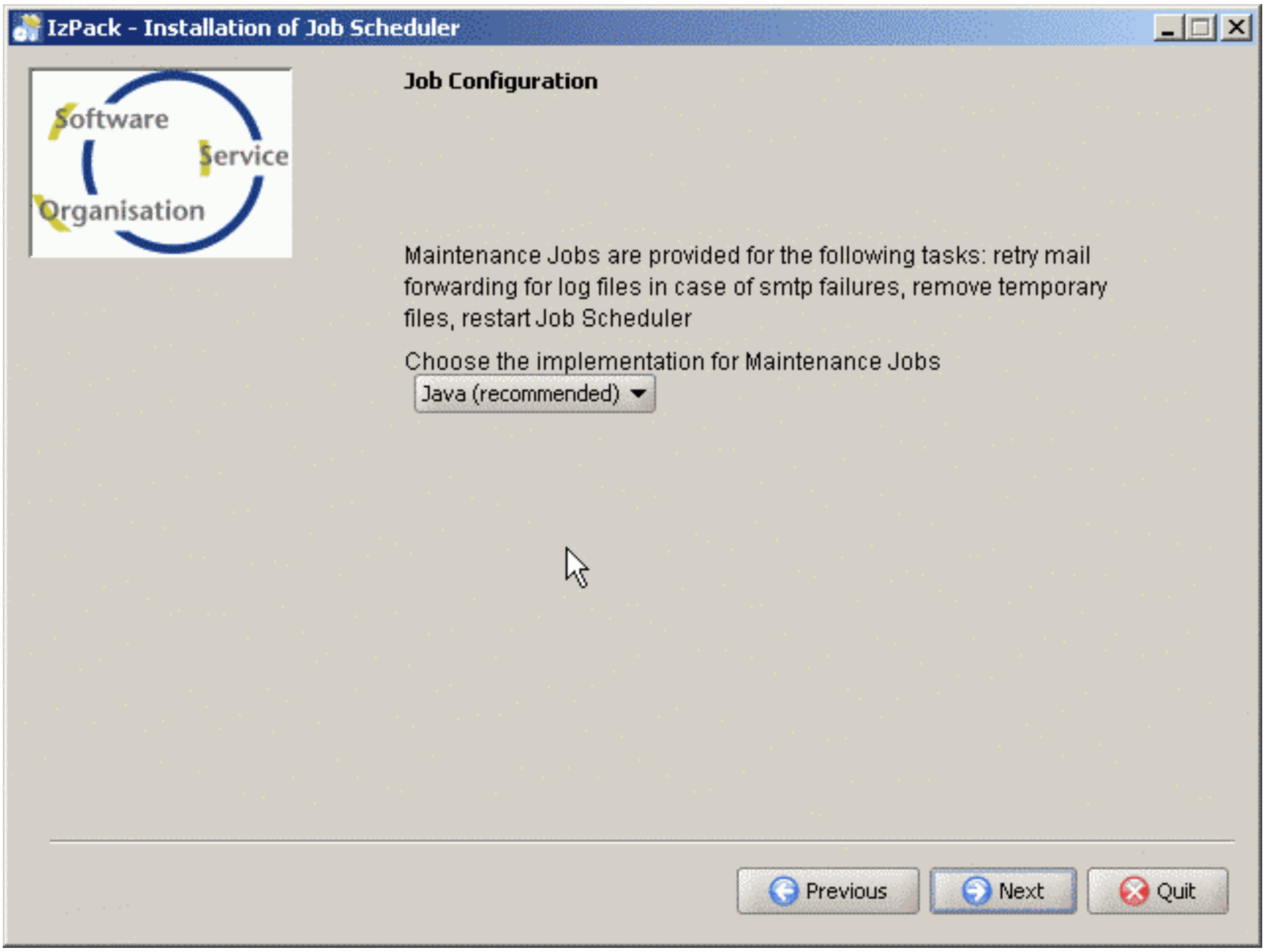
This dialog form is only relevant for MySQL and SQL Server databases and can be omitted when Oracle, PostgreSQL, Firebird or DB2 are being used. The script for the creation of the database tables is started by the setup program and requires a JDBC driver appropriate to the database system being used. The drivers for Oracle, PostgreSQL, Firebird and DB2 are included in the setup. However, because of licensing restrictions, the relevant MySQL and SQL Server JDBC driver must be manually specified here.

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 30).

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

Maintenance 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 Maintenance Jobs

Java (recommended)

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.

Language	*Housekeeping Job*
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

The documentation for these jobs is to be found in HTML format after installation of the Job Server in the *[install_path]/jobs* folder.

The entries made in this form are saved in the *[install_path]/config/scheduler.xml* file, where they can be later on changed manually (page 30).

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 which can be selected as a package during the setup.

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

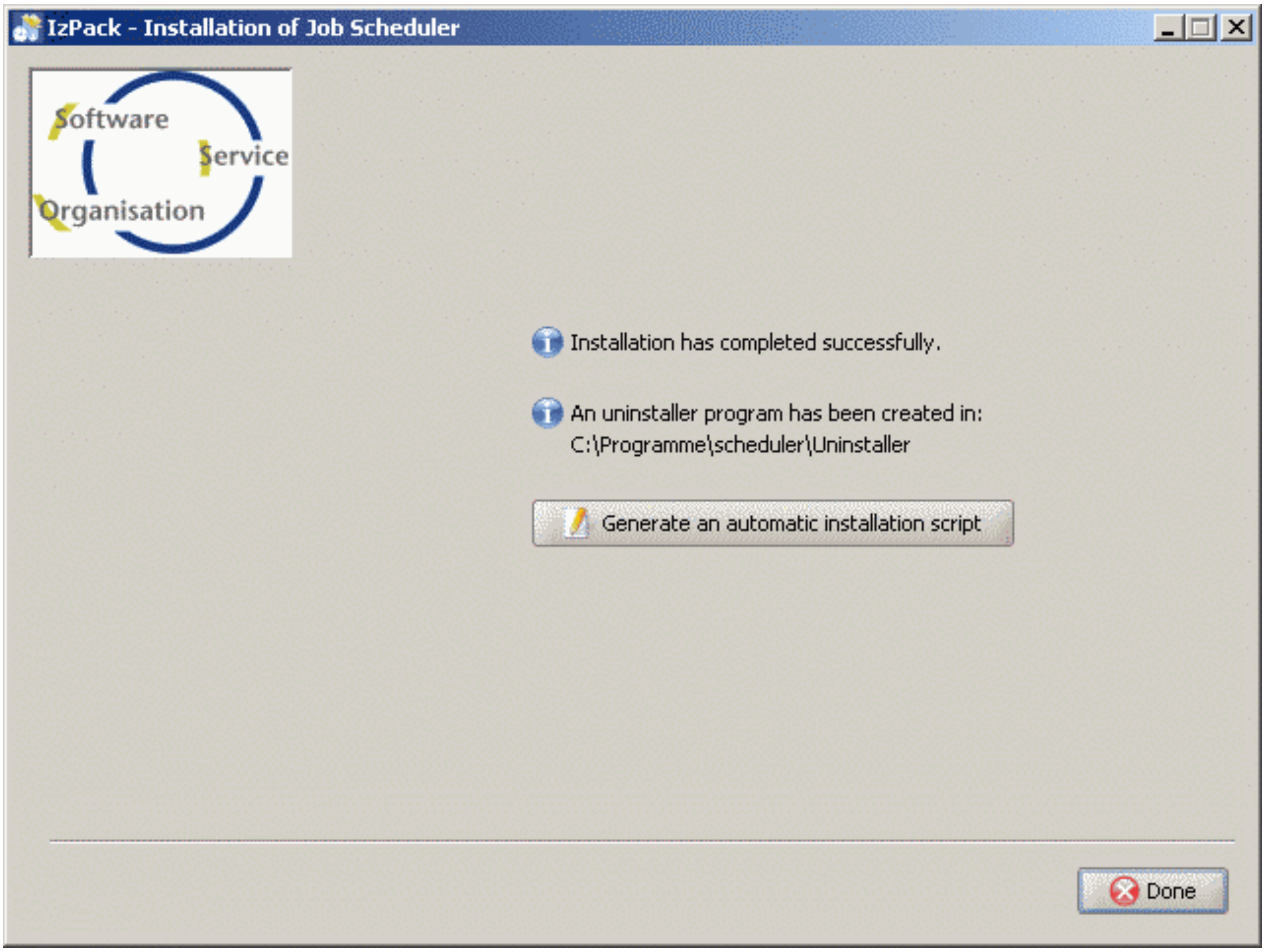
+ **bin** (Windows)
  - **hostjava.dll** Program library
  - **hostole.dll** Program library
  - **jobeditor.cmd** Start script for the Job Configuration Editor
  - **jobscheduler.cmd** Start script for the Job Scheduler
  - **managedJobChainExport.cmd** Export script for Managed Jobs
  - **managedJobChainImport.cmd** 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** Program library
+ **bin** (Linux/Solaris)
  - **jobeditor.sh** Start script for the Job Configuration Editor
  - **jobscheduler.sh** Start script for the Job Scheduler
  - **managedJobChainExport.sh** Export script for Managed Jobs
  - **managedJobChainImport.sh** Import script for Managed Jobs
  - **scheduler** Job Scheduler executable 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**
  + **html** Job Scheduler web interface
  - **factory.ini** runtime configuration file
  - **scheduler.xml** XML configuration file
  - **scheduler.xsd** Schema definition for XML configuration files
  - **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_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
    + **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 the 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 the mssql directory)
    + **fbsql** Firebird 1.5
        - ***.sql** (see the mssql directory)
    + **pgsql** PostgreSQL 8.x
        - ***.sql** (see the mssql directory)
        - **sos.sql** (Managed Jobs)
    + **db2** IBM DB2 8.x
        - ***.sql** (see the pgsql 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** libraries (Linux/Solaris)
+ **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.



IzPack - Installation of Job Scheduler
Software
Service
Organisation

Installation has completed successfully.

An uninstaller program has been created in:
C:\Programme\scheduler\Uninstaller

Generate an automatic installation script
Done

Instructions for starting the automatic setup script can be found under Batch Installation (page 21).

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 and Oracle database systems are supported. The Job Scheduler setup program creates the necessary database tables if the *Database Support (page 5)* package is installed and the database connection is specified in the respective setup form.

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

1.7.1 Manual Database Table Creation

SQL scripts which create the database tables required by the Job Scheduler are available, should the tables not be correctly created by the setup program. To make up for this call the script `[install_path]/install/scheduler_install_tables.(sh|cmd)`.

Please 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 30).

1.8 Starting and stopping the Job Scheduler

1.8.1 Job Scheduler Demon for Unix

For Unix 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
```

Besides 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 to the appropriate startup/shutdown directory, usually this is /etc/init.d.

After setup the Job Scheduler is not automatically started, please use the above script.

1.8.2 Job Scheduler Service for Windows

For Windows the Job Scheduler is installed as service. You can find it by opening your service panel and looking for a service name that starts with "SOS Job Scheduler".

To start the Job Scheduler manually please ensure that the service is not started and use the following script:

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

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

Besides 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 parametrized 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 parametrized 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>
       <installpath>./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"/-->
       </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=""/>
           <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=""/>
        </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"/>
        </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>
            <!-- Configuration for JDBC Driver -->
            <entry key="connector" 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 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	The <selected> element posesses elements with the form <pack index="[integer]"/>. The coresponding set-up package will be installed for each <pack> element selected. • `<pack index="0"/>` is the basic *Job Scheduler* package. **This package must always be specificied**. • `<pack index="1"/>` is the *Update Service* package. • `<pack index="2"/>` is the *Database Support* package. • `<pack index="3"/>` is the *Web Interface* package. • `<pack index="4"/>` is the *Managed Jobs* 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 *Housekeeping Jobs* package. • `<pack index="6"/>` is the *Sample Jobs* package. • `<pack index="7"/>` is the *MySQL Maintenance Jobs* package. A pre-requisite for the selection of this package is the specification of `<pack index="2"/>`. More information about the packages and the conditions of their use can be found under Setup Packages (page 5). It is extremely important that the dependencies between the packages described above are observed.
Network Configuration	
value of key= "serviceHost"	The host name or IP address, on which the *Job Scheduler* is to be operated.
value of key= "serviceId"	The specification of a Scheduler ID makes sense, when more than one *Job Scheduler* is to be operated.
value of key= "serviceAllowedHost"	It is recommended that TCP access is allowed for the 'localhost', other host names or IP adresses 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 *Job Scheduler* 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	

Element/Attribute	*Description*
value of key= "clusterOptions"	The *Job Scheduler* can be operated independantly of any other *Job Schedulers*; as the primary *Job Scheduler* on 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 *Job Scheduler* is to be operated without a database or independantly. The value '-exclusive' should be specified when the *Job Scheduler* is to operate as the primary in a cluster. The value '-exclusive -backup -backup-precedence=[N]'should be specified if the *Job Scheduler* is to operate as a backup in a cluster, where [N] is an integer (see Installation of a Backup Cluster (page 27)).
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.
value of key= "mailTo"	The recipient(s) of log file mails. The addresses of multiple recipients should be seperated by commas.
value of key= "mailCc"	The carbon copy recipient(s) of log file mails. Multiple recipients should be seperated by commas.
value of key= "mailBcc"	The blind carbon copy recipient(s) of log file mails. The addresses of multiple recipients should be seperated by commas.
Update Service Configuration The following settings will only be evaluated, when the `<pack index="1"/>` element exists.	
value of key="check ForUpdateStarttime"	The *Job Scheduler* 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="check ForUpdateStartday"	The *Job Scheduler* 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 *Job Scheduler* be available, then an appropriate e-mail will be sent to the administrator. In addition, the *Job Scheduler* can start an automatic download of the release. Possible values for this parameter are 1 (automatic download) or 0 (no automatic download).

Element/Attribute	*Description*
Job-Konfiguration The following settings will only be evaluated, when the `<pack index="5"/>` element exists.	
value of key="job AutomationLanguage"	Standard-Jobs zur Automation werden für folgende Aufgaben ausgeliefert: Wiederholter eMail-Versand von Protokollen nach Ausfall des Mail-Servers, Entfernen temporärer Dateien, Neustarten des *Job Scheduler*s. Wählen Sie eine Implementierung der Standard-Jobs. Mögliche Werte sind 'scheduler_automation_java.xml' (empfohlen), 'scheduler_automation_javascript.xml', 'scheduler_automation_vbscript.xml' und 'scheduler_automation_perl.xml'.
Database Configuration The following settings will only be evaluated, when the `<pack index="2"/>` element exists.	
value of key= "databaseDbms"	Specifies the database system. Possible values are: 'mysql' (for MySQL), 'oracle' (for Oracle), 'mssql' (for MS SQL Server), 'pgsql' (for PostgreSQL), 'fbsql' (for Firebird) and 'db2' (for DB2).
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: MySQL 3306, Oracle 1521, SQL-Server 1433, PostgreSQL 5432, Firebird 3050, DB2 50000.
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.
Configuration of the JDBC Driver The following settings will only be evaluated, when the `<pack index="2"/>` element exists.	
value von key= "connector"	This parameter need only be set when either a MySQL or a MS SQL Server database has been selected. In this case, the appropriate JDBC driver must be installed. (The JDBC drivers for MySQL or MS SQL Server cannot be included in the *Job Scheduler* installation program because of licensing reasons.) The path to the JDBC driver must be specified here (e.g. mysql-connector-java-*.jar für MySQL, sqljdbc.jar für MS SQL Server).

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 is (fully) recovered when reinstalling the Job Scheduler.

With the exception of the Scheduler ID, the three forms of the basic package *Job Scheduler* (page 9) can be m issed out. Should changes be required in the *network* and / or *e-mail configuration*, then these must be carried out manually (page 30).

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 is particularly important 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 30) file. The host and port of the *main_scheduler* should then be entered in the *[install_path]/web/custom/custom.inc.php* (page 30) 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). 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 21). 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*.

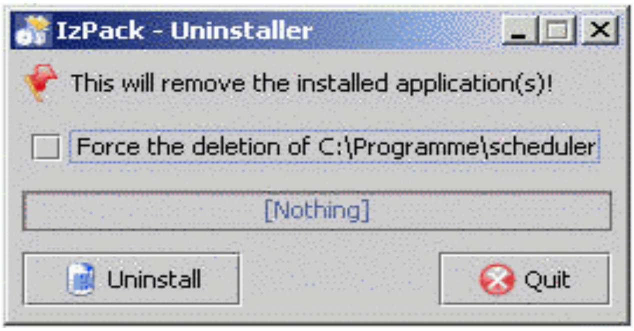
5 Deinstallation

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 that the removal of the Job Scheduler be confirmed.



IzPack - Uninstaller

This will remove the installed application(s)!

□ Force the deletion of C:\Programme\scheduler

[Nothing]

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 Linux/Solaris 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 can 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 have been deactivated, then a renewed installation of a Job Scheduler with the same [scheduler_id] is 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. The path to the Job Scheduler installation directory is denoted with [install_path].

- **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 Linux/Solaris

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 Server 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.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 classpath of the Java archives 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 File

The *scheduler.xml* file is in the *[install_path]/config* directory. The host und port information of the Job Scheduler are to be found here, 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.

6.3 The jobscheduler.sh File (for Unix)

The *jobscheduler.sh* file is relevant only for unix and is 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.

6.4 The custom.inc.php File

The file *custom.inc.php* is to be found in the *[install_path]/web/custom* directory, should the *Web Interface (page 5)* package have been installed during setup. This file is used to specify database connection information; the Job Scheduler language, host and port 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 no value have been given for '[port]', then the standard port used by the database will be used. Should the value for '-host' not be given, then 'localhost' and the standard port will be used.

Database Type
Oracle, MySQL, Microsoft SQL Server, PostgreSQL, DB2, Firebird and ODBC data sources are supported. The database type is set using the PHP constant *APP_CONNECTION_CLASS* 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 Firebird:
  ```
  if(!defined('APP_CONNECTION_CLASS')) {
     define ( 'APP_CONNECTION_CLASS', 'sos_fbsql_record_connection' ); }
  ```
- For DB2:
  ```
  if(!defined('APP_CONNECTION_CLASS')) {
     define ( 'APP_CONNECTION_CLASS', 'sos_db2_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' ); }
```

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 5)* package during setup requires that the web server 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.

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)
- 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)

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]
```

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 is 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 *./config/.version* file. After the update has been downloaded, the directory in which it is save 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 Operating with 64bit

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

- Job Scheduler is a 32bit program
  The Job Scheduler is implemented as a 32bit executable. You could ask why is this software has not been ported to 64bit? The simple answer is: there is no need for this.
  - You can run the 32bit Job Scheduler executable on every supported 64bit platform (Windows, Linux, Solaris, HP-UX PA-RISC, HP-UX Itanium). This is independent from the fact that you possibly schedule 64bit programs by job implementations on a 64bit platform as the Job Scheduler creates platform specific processes.
  - The possible difference in execution speed between a 32bit and a 64bit Job Scheduler executable would be neglectable. Most of the execution time is consumed by jobs and not by the Job Scheduler, there is no benefit in execution speed.
  - Modifying the Job Scheduler code for 64bit support would require to fork the code. 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 during a longer period of time which would slow down the current roadmap. We estimate the benefit of new features higher than the benefit of a simplified installation by a native 64bit Job Scheduler.
- Java 32bit JRE
  A Java 32bit JRE is required for executing the setup and to operate the Job Scheduler.
  To our experience it's a good idea to use a JRE provided by SUN as JREs from other distributions, e.g. IBM Websphere, tend to be somewhat incomplete or modified for individual products to an extent that they are not sufficiently compatible to standards.
  - Windows 64bit
    You could use the JRE 1.5 that is included in the resp. Job Scheduler download or install a separate 32bit JRE. You should not include the newly installed 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 could be completed by 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* as in:
       `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 as in:
       `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 configuration file *.\config\sos.ini* like this:
       ```
       [java]
       vm = C:/Program Files (x86)/Java/jre1.5.0_01/bin/client/jvm.dll
       ```
  - Unix/Linux 64bit
    You have to download a 32bit JRE should this version not be available on your machine.
    1. Download and install the 32bit JRE.
    2. Temporarily add the location of the newly installed JRE to your *PATH* as in:
       `PATH=$HOME/jdk1.5.0_06/jre/bin:$PATH`
    3. Unzip the Job Scheduler download archive to a temporary directory, e.g. `/tmp/scheduler` as in:
       ```
       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 as in:
       `java -jar /tmp/scheduler/scheduler_linux32.jar`

5. Having completed the setup, add the location of the Virtual Machine to to the environment variable *LD_LIBRARY_PATH* in the startup script `./bin/jobscheduler.sh` like this:
   ```
   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 platform
  Please, make sure that you use a 32bit Java JRE when executing the setup. Depending from the target platform there are differences in installing the 32bit Job Scheduler for a 64bit platform:
  - Windows 64bit
    As for every 32bit program you have to install the Job Scheduler and Java JRE in the folder *C:\Program Files (x86)* (or language specific folder name) as this folder is reserved by the operating system for 32bit programs. Installations to different folders will not work.
    Should this not be a perfect location for your Job Scheduler configuration files and log files then you could adjust the path for these files in the script *.\bin\jobscheduler.cmd* by changing the respective startup options (*-config*, *-log-dir*, *-log* etc.). To apply such changes the Windows service has to be re-installed, you could use the following commands (this sample uses *C:\Program Files (x86)\scheduler* as your installation directory):
    ```
    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/Linux 64bit
    There is no difference in installing the 32bit Job Scheduler for this platform. However, changes to the startup script might be required by the Java JRE location (see above).

9 Troubleshooting

In addition to the following list further issues are covered by the FAQ in the Job Scheduler web site.

Choose the appropriate JDBC Driver
The JDBC Driver must correspond to the version of your database system. Should problems occur in the Job Scheduler concerning database connections, then the following error might be displayed in the log files:
`Error SOCKET-61 ECONNREFUSED Connection refused (TCP-Port not available) [connect].`
For legal reasons we cannot give any recommandations, but we can tell you from our experiences:

- MySQL
  For MySQL version 4.x a good practice is using `mysql-connector-java-3.1.8-bin.jar`.
- PostgreSQL
  we use the PostgreSQL 7 JDBC Driver in the setup as version 8 currently does not support to create procedures via JDBC. The version 8 JDBC Driver is used for operation of the Job Scheduler, currently we are not aware of major errors in this driver version.
- SQL Server
  For SQL Server the JDBC Driver `sqljdbc.jar` can be used with both versions 2000 and 2005. However, an older JDBC Driver version might not work with SQL Server 2005.
- Oracle
  we are not aware of any major problems of the Job Scheduler running with the JDBC Driver ojdbc14.jar for Oracle 9.2 and 10g.

ANSI-Mode in MySQL
While 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 quite a bunch of database systems and use ANSI compliant SQL. Switching automatically to ANSI mode does not work with older MySQL versions 4.0.x. You have to set this mode yourself in the database server. Open the file `my.cnf` and insert in the section `[mysqld]` an entry `SQL_MODE=ANSI_QUOTES` or add the parameter `--ansi` to your MySQL start script. You have to restart the database server after changes to this configuration.

Database connections to MySQL get lost
You might 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 the connection to your MySQL database was idle for some hours without any jobs running, then MySQL will close the connection without telling this to the client, i.e. the Job Scheduler. To change this behaviour you can change the value of the system variable `wait_timeout`. This value assigns the maximum duration in seconds of non-interactive idle connections to the database.
Alternatively you could run a job like `scheduler_dequeue_mail` that is more frequently repeated; this job dequeues mails that have previously been stored in case of a failure of your 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 use an older version of the JDBC Driver, (e.g. `msbase.jar`, `mssqlserver.jar`, `msutil.jar`), then the URL for the JDBC connection in the configuration file `./config/factory.ini` is different from the newer version `sqljdbc.jar`.
.
Older version:

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

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

eMails cannot be sent

The sender address must be a valid address for your mail server. Problems might occur with outgoing mails if the domain part of this address is not valid for the mail server. Adjust the sender address in the entry `log_mail_from` of the configuration file `./config/factory.ini`.

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.

For normal operation a restart is not required, this applies as well for longer periods.

The Job Scheduler asks for a licence key, isn't this software Open Source?

Running the Job Scheduler on Unix with the binary file `./bin/scheduler` or on Windows with the file `./bin/scheduler.exe` will result in the 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 shell script `./bin/jobscheduler.sh` (Unix) or `./bin/jobscheduler.cmd` (Windows). The binary files are parametrised by the start script. One of the parameters (`-sos.ini=...`) in the start script addresses the licence file `./config/sos.ini`, that contains a free licence key for the GPL version of the *Job Scheduler*.

There is no difference between GPL and commercially supported versions, in fact this is a technical licence key that helps us to identify commercial customers in case of support incidents.

Index