



Job Scheduling

JOB SCHEDULER

Installation and Configuration

March 2006

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: March 2006

Table of Contents

1 Installation	4
1.1 Installation Using the Setup Program	4
1.2 Setup Packages	5
1.3 The Sample Jobs Package	7
1.4 Setup Forms	7
1.4.1 The Basic Job Scheduler Forms	8
1.4.2 The Database Support Package Forms	11
1.4.3 The Housekeeping Jobs Package Form	13
1.5 Directory Structure after Installation	14
1.6 Automatic Installation	16
1.7 Database Configuration	16
1.7.1 Manual Database Table Creation	17
1.8 Starting and stopping the Job Scheduler	17
1.8.1 Job Scheduler Demon for Unix	17
1.8.2 Job Scheduler Service for Windows	17
2 Multiple Installation	18
2.1 Reinstallation of the Job Scheduler	18
2.2 Installation Alongside an Already Existing Installation	18
3 Deinstallation	19
3.1 Removal Using the Uninstaller	19
3.2 Manual Removal on Windows	20
3.3 Manual Removal on Linux/Solaris	20
4 Configuration	21
4.1 The factory.ini File	21
4.2 The scheduler.xml and scheduler.xsd File	21
4.3 The jobscheduler.sh File (for Unix)	21
4.4 The custom.inc.php File	21
4.5 Configuration of the Web Server	23
Index	24

1 Installation

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

- **Database Configuration** (page 16) (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 Web Interface of the Job Scheduler allows retrieval of past job protocols. Further, when database support is not selected, the choice of additional packages which can be installed alongside the Job Scheduler is restricted. MySQL, Oracle, Microsoft SQL Server, PostgreSQL, Firebird and DB2 database systems are supported by the Job Scheduler.

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

- **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 23) (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 "*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 selected. Further details of the Job Scheduler configuration are to be found in the Setup Forms (page 7) chapter.

After selection of the required packages, the necessary files are copied into the installation directory and the scripts are executed that configure the installation packages. The processing of 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.

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" %*
```

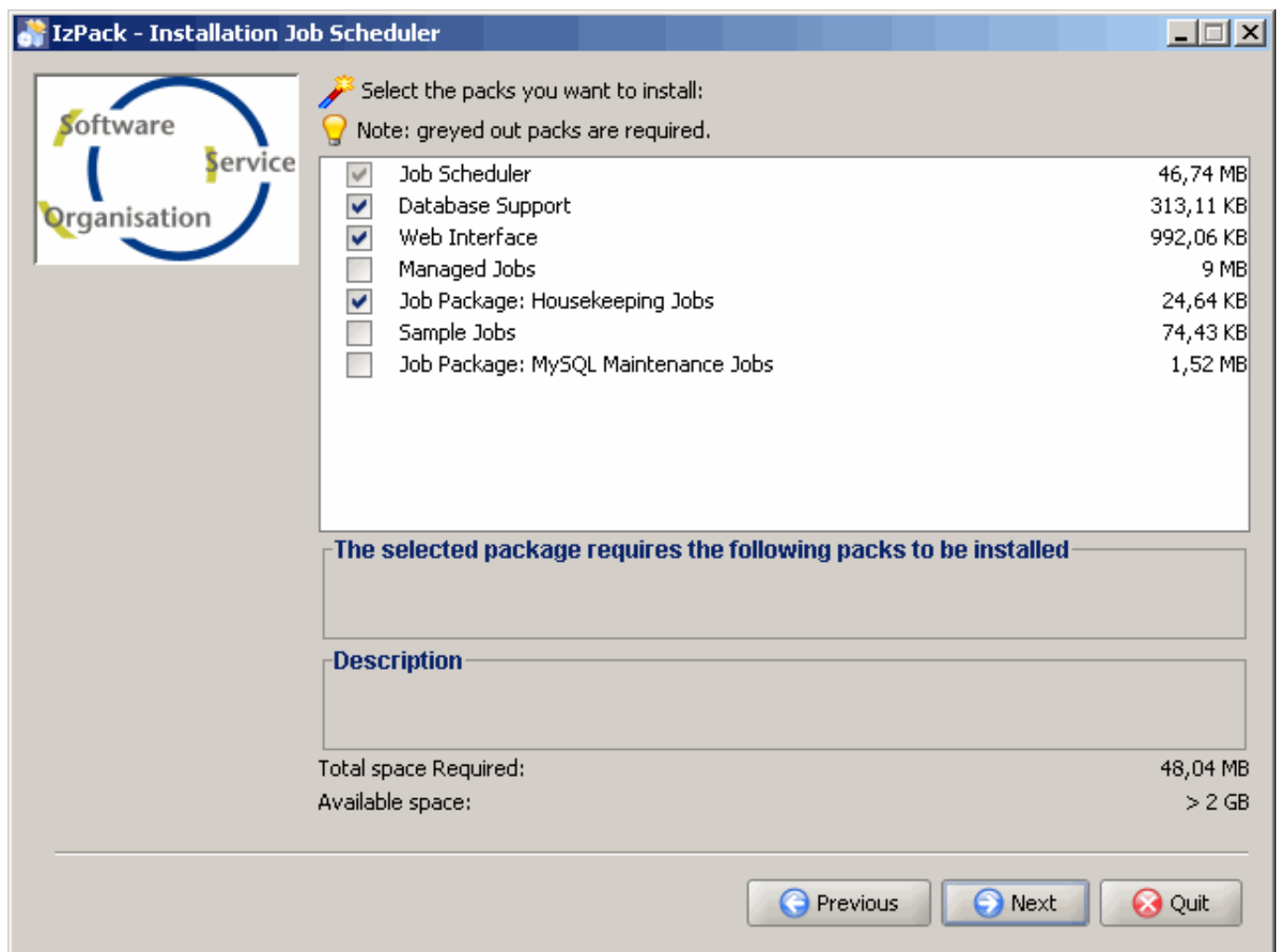
1.2 Setup Packages

The following packages may be selected during setup:

- **Job Scheduler**
This is the basic package and must be installed.
- **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.
- **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).
- **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.



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>

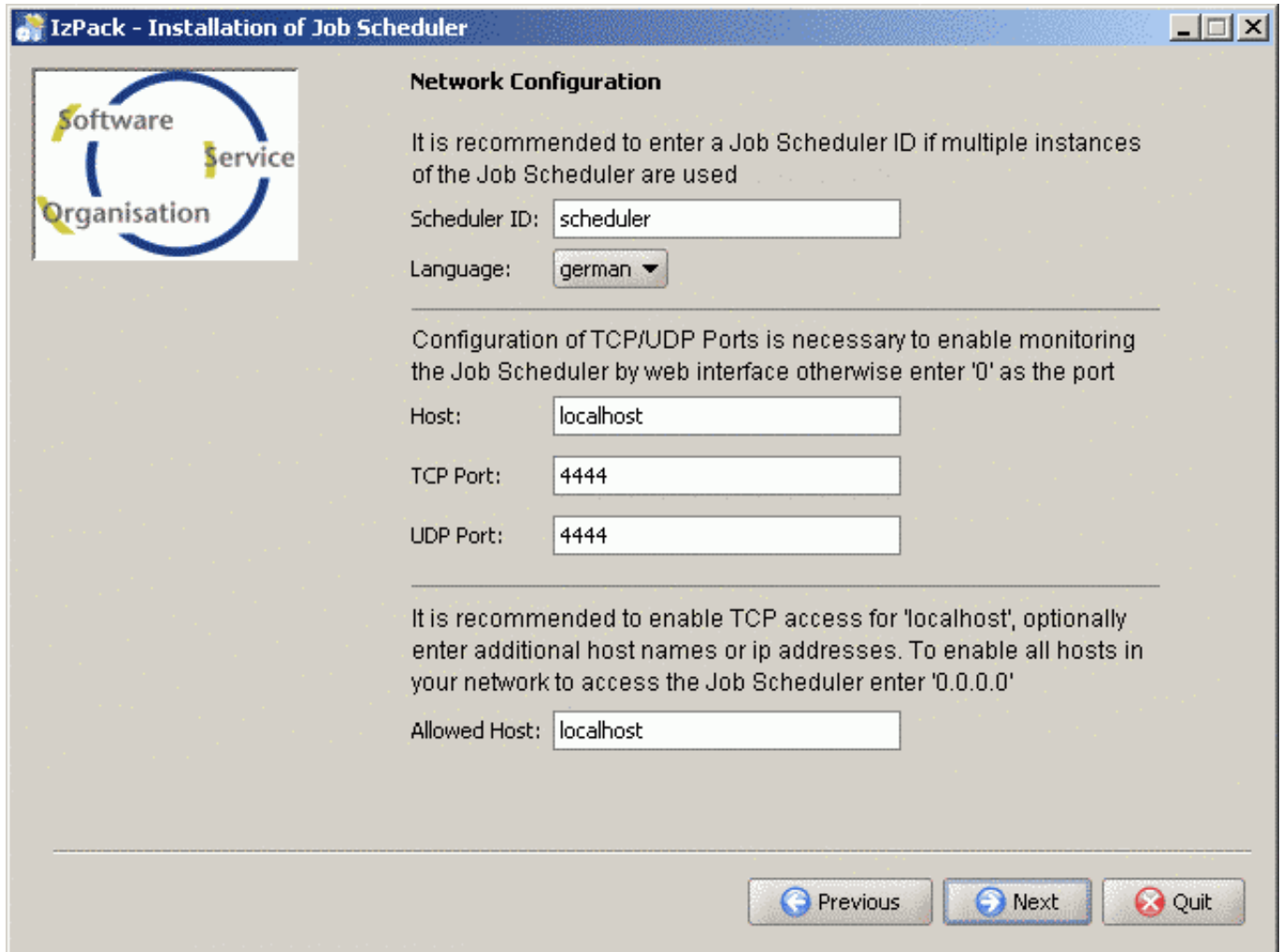
</spooler>
```

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 installed.

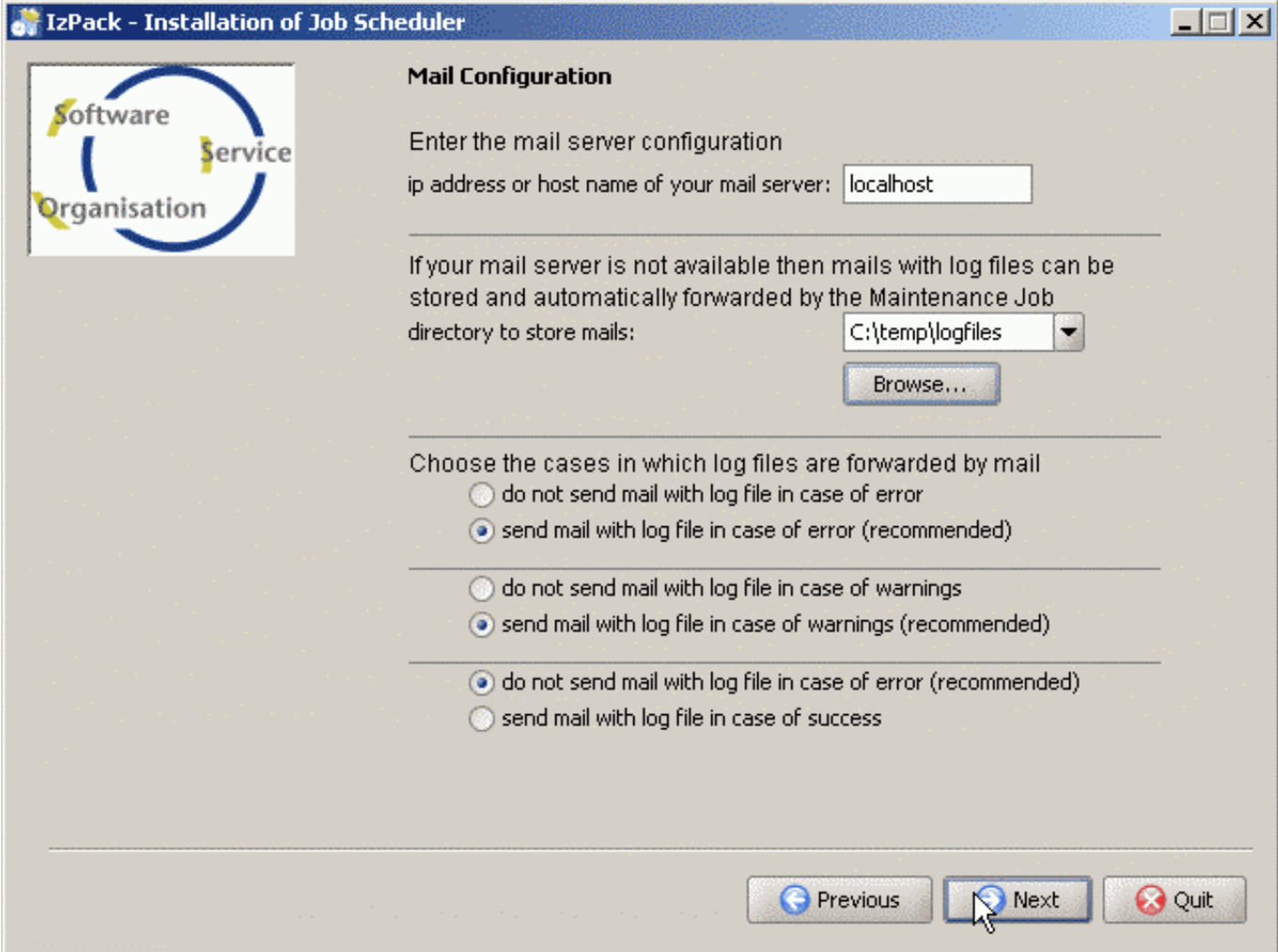
1.4.1 The Basic Job Scheduler Forms



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 form *sos_scheduler_[scheduler_id]*.

The language of the PHP user interface is determined in the selection box situated underneath the ID input box. Note that this selection is relevant only when the *Web Interface* package has been installed. 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. Entries made for language, 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 written to the *[install_path]/config/scheduler.xml* file. Both configuration files can be changed manually (page 21) later on.

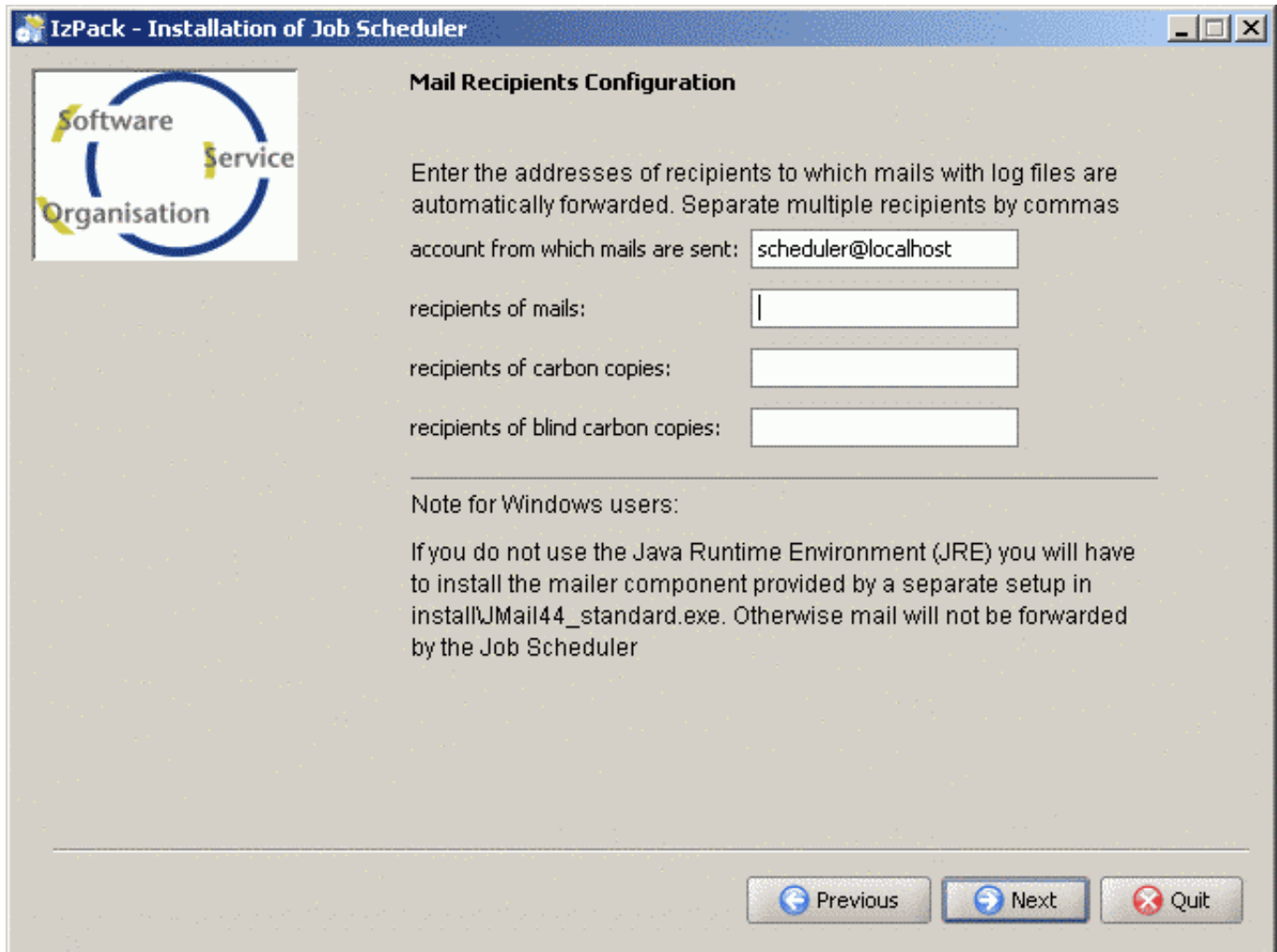


The screenshot shows a Windows-style window titled "IzPack - Installation of Job Scheduler". On the left is a logo for "Software Service Organisation". The main area is titled "Mail Configuration" and contains the following elements:

- Text: "Enter the mail server configuration"
- Text input field: "ip address or host name of your mail server:" with the value "localhost" entered.
- Text: "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:"
- Text input field: "C:\\temp\\logfiles" with a dropdown arrow.
- Button: "Browse..."
- Text: "Choose the cases in which log files are forwarded by mail"
- Radio button options:
 - ☐ 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
- Navigation buttons at the bottom: "Previous", "Next" (highlighted with a mouse cursor), and "Quit".

The SMTP Server is specified here along with information regarding whether the Job Scheduler should automatically forward job protocols by e-Mail.

The values entered here configure the *[install_path]/config/factory.ini* file, which can also be changed manually (page 21) 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:

recipients of mails:

recipients of carbon copies:

recipients of blind carbon copies:

Note for Windows users:

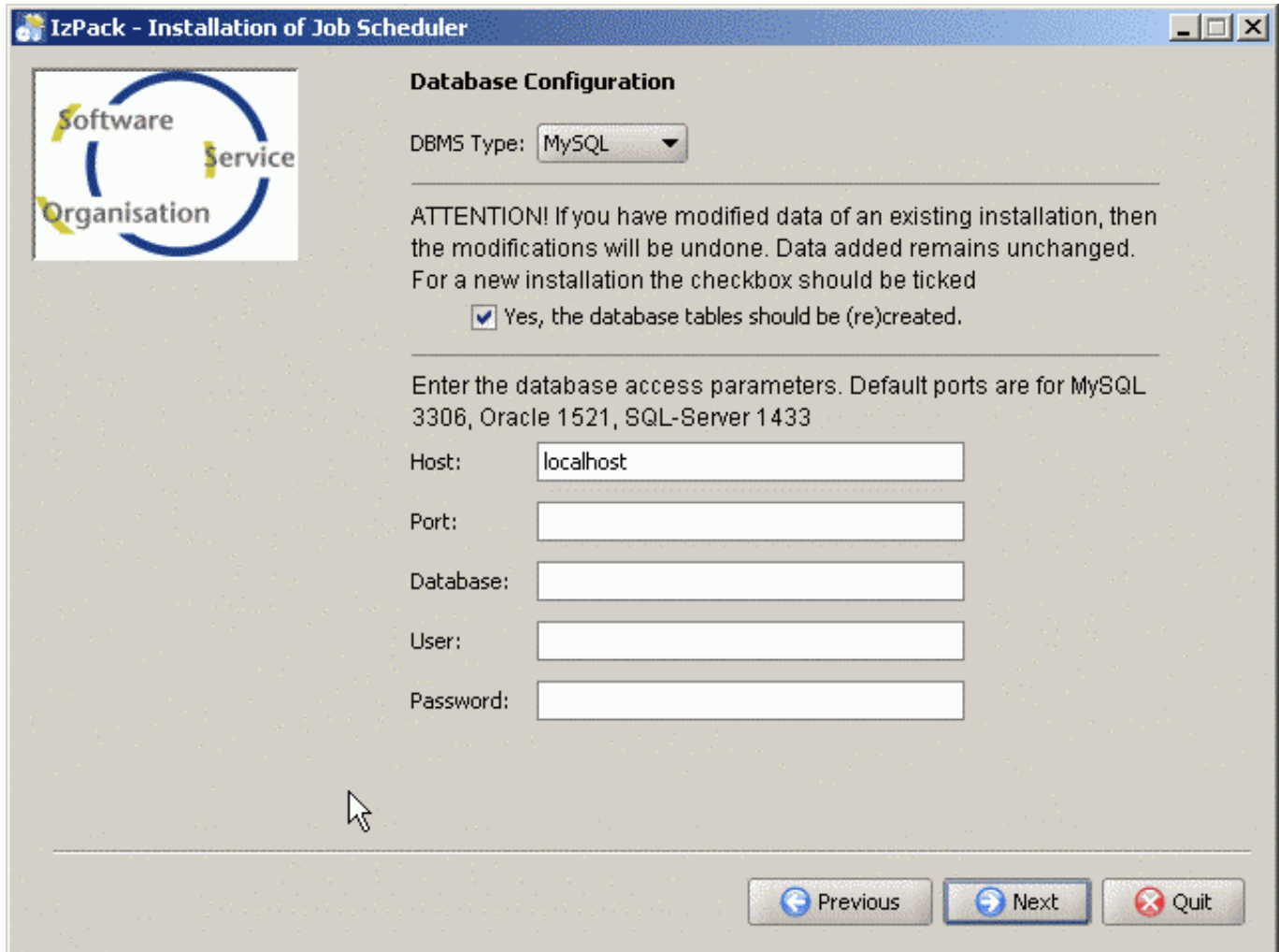
If you do not use the Java Runtime Environment (JRE) you will have to install the mailer component provided by a separate setup in `installJMail44_standard.exe`. Otherwise mail will not be forwarded by the Job Scheduler

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 separated by commas.

In order to use the Job Scheduler 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 21).

1.4.2 The Database Support Package Forms



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:

Port:

Database:

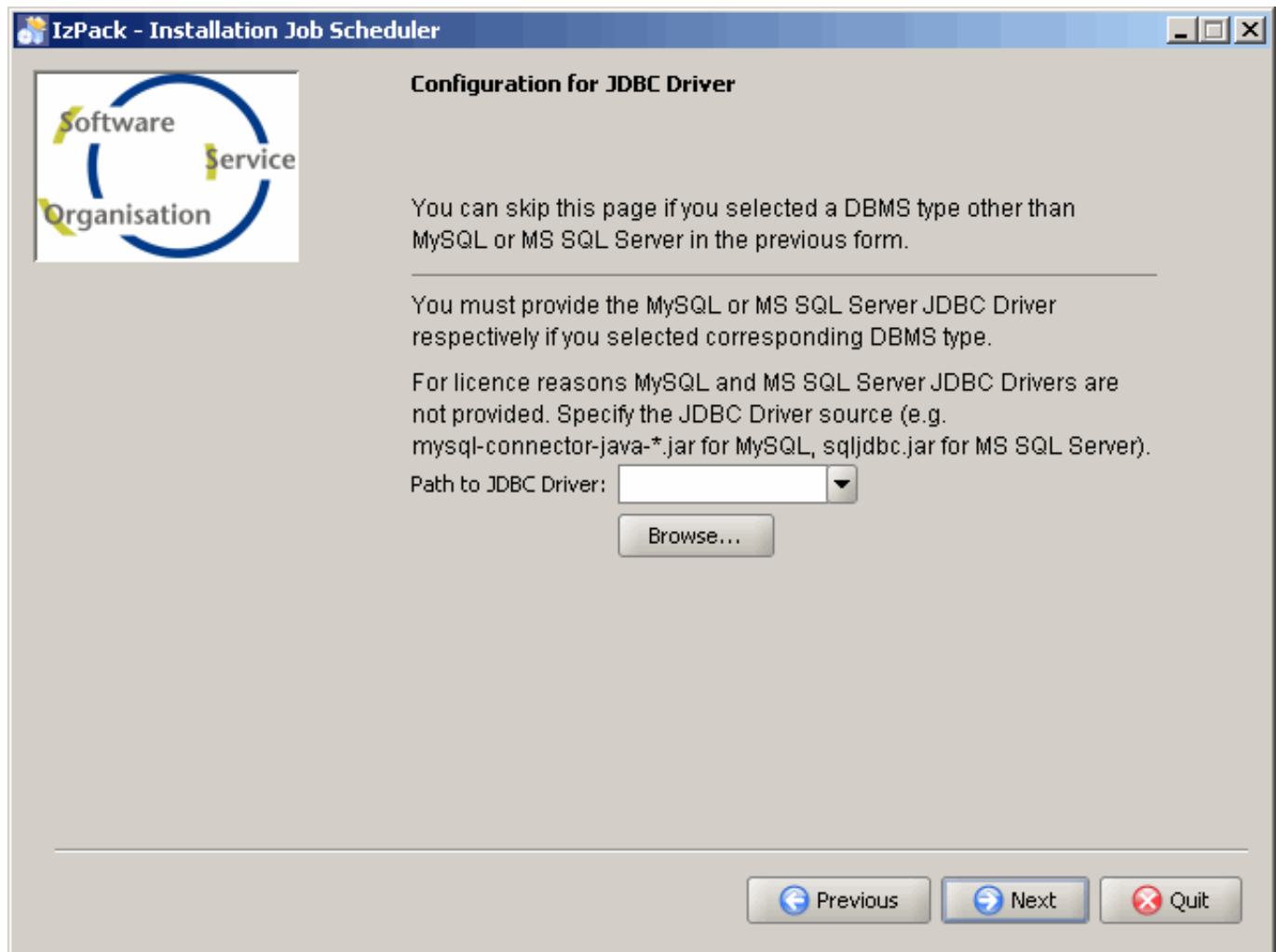
User:

Password:

The database system is specified in the upper selection box on this form and the database connection information 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 17).

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 21) if required.

If the database system is Firebird then you have to ensure that no other connections to the database server exist during installation.



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

As this driver will also be required by the Job Scheduler later, 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 21).

1.4.3 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) ▼

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. 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 21).

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.

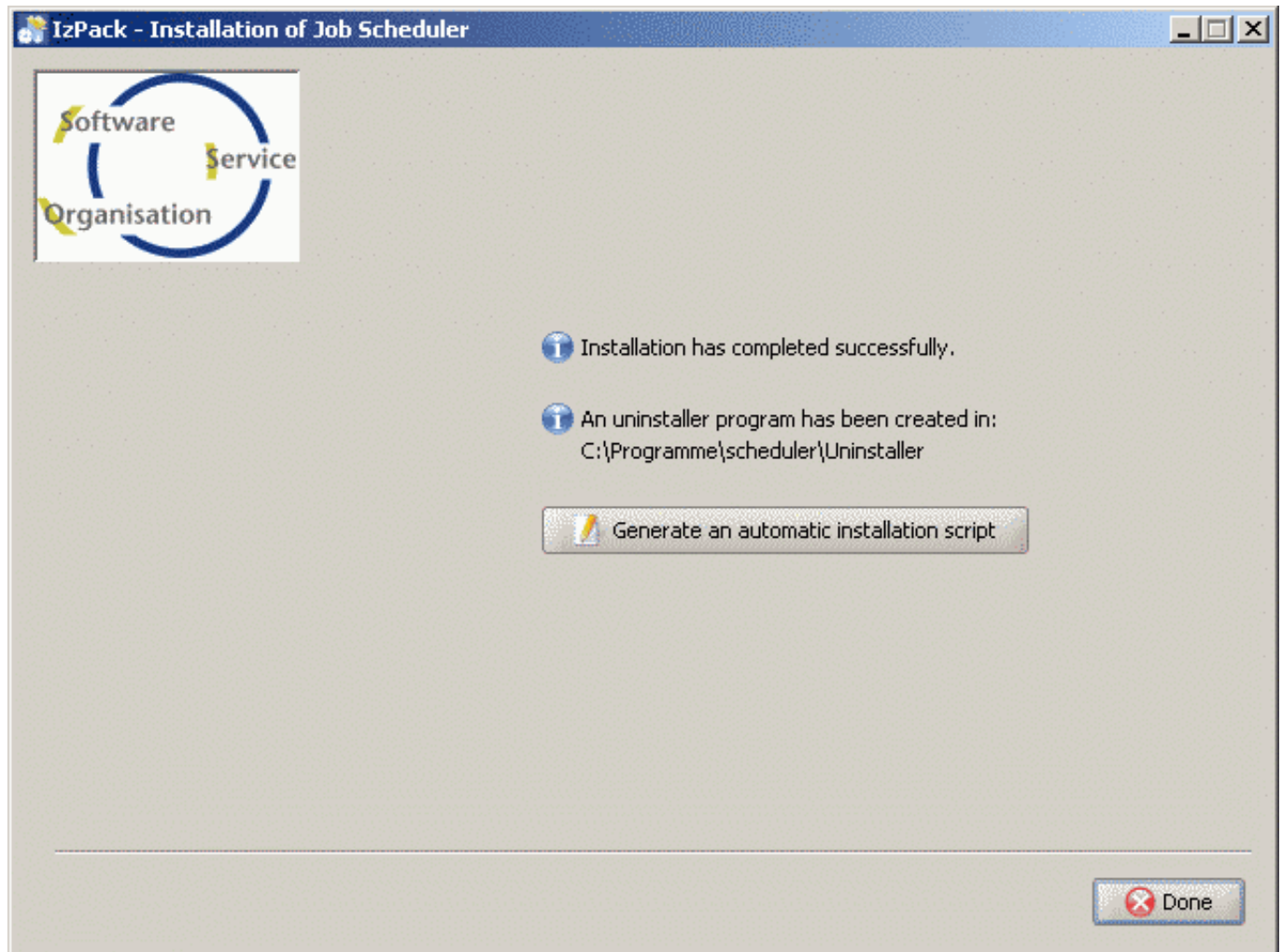
The following directory structure should be found in the Job Scheduler [install_path]:

- + **bin** (Windows)
 - **hostjava.dll** Program library
 - **hostole.dll** Program library
 - **jobscheduler.cmd** Start script
 - **managedJobChainExport.cmd** export Script for job packages
 - **managedJobChainImport.cmd** Import script for job packages
 - **scheduler.exe** Job Scheduler binary
 - **scheduler.exe.local**
 - **settingsImport.cmd** import script for database settings
 - **spidermonkey.dll** Program library
- + **bin** (Linux/Solaris)
 - **jobscheduler.sh** Start script
 - **managedJobChainExport.sh**
 - **managedJobChainImport.sh**
 - **scheduler** Job Scheduler binary
 - **scheduler_safe.sh**
 - **settingsImport.sh**
 - **setuid**
- + **config**
 - + **html** Job Scheduler web interface
 - **factory.ini**
 - **scheduler.xml**
 - **scheduler.xsd**
 - **scheduler_mail.xsl**
 - **sos.ini** licence file
 - **sos_settings.ini** database connection for shell scripts
 - **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 setup has been completed, it is possible to save an XML file to be 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.



The automatic setup script is started as follows:

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

Note that [setup.jar] is the Setup program (page 4) for the operating system being used and [auto_install.xml] the automatic installation script.

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` and `[install_path]/web/custom/custom.inc.php` configuration files (page 21).

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, cancel 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 Multiple Installation

2.1 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 specification of the PHP interface language, the three forms of the basic package *Job Scheduler* (page 8) can be missed out. Should changes be required in the *network* and / or *e-mail configuration*, then these must be carried out manually (page 21).

The *database configuration* (page 11) 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 specify the JDBC driver in the *Database Configuration* form once more.

2.2 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 8) 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 21) file. The host and port of the *main_scheduler* should then be entered in the *[install_path]/web/custom/custom.inc.php* (page 21) file manually.

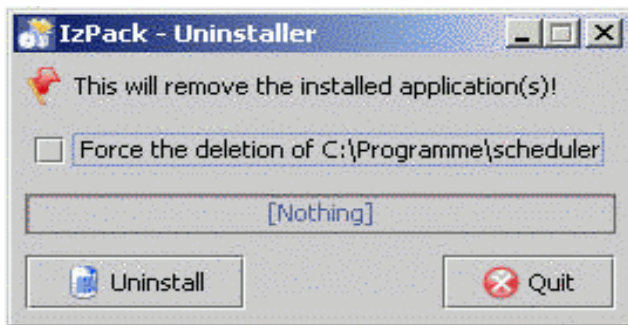
3 Deinstallation

3.1 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.



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.

3.2 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]`

3.3 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]`

4 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.

4.1 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.

4.2 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.

4.3 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.

4.4 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' ); }
```

4.5 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.

Index

C

custom.inc.php 8, 11, 15, 18, 21

D

Database 4, 5, 6, 11, 16, 19, 20

F

factory.ini 9, 10, 11, 12, 14, 21

J

jobscheduler.sh 21

S

scheduler.xml 8, 13, 14, 18, 21

U

Uninstaller 15, 19

W

Web Server 4, 15, 19, 20, 23