OnNubis

BIS

Scheduler

Architecture, Install and Setup

For Unix/Linux and Windows

Admin Manual

English

May 2018

Published by OnNubis

Contents

Introduction

BIS Scheduler

Goal:

It allows the Navigator and Executives to run reports in a disconnected mode at a specified date and time and defines which are the groups and users that can access the results of the report execution. It also possible to generate the output as XML file, stored in a predefined server folder/directory.

Advantages:

The same report is executed once in the DB Server and consumed by several users. Long running reports can be launched during a period when the DB Server is less used, lowering the impact of long running queries and optimizing the Server CPU and I/O usage.
The new results can be uploaded to the client machines faster and the users can transform them as desired.
After login in the Navigators and Executives are notified if there are new cached reports generated, since the last user logout.

Users' functions:

Administrator:

- Defines the Scheduler sleep time and can Stop the Scheduler.
  By default the sleep time is 30 seconds.
- Defines the Days/Hours window when is possible to run the scheduled tasks.
- Can remove a task from the scheduler
- Can reset the status of a task

Navigator and Executive:

- Submits reports, defines who can access them, the frequency of the reports' results refresh.
- As the owner of a submited task he can also:
  1) remove a Scheduled report
  2) Change the frequency of the Scheduled report
  3) Change the output type of the report

Bis-for-Excel:

- Consumes the cached reports, generated by the Scheduler.

Architecture

The S4Scheduler is based on Java technology, with the purpose to run in a server, scheduled reports submitted by the BIS-Navigator and the BIS-Executive.

The server components of the s4Scheduler is made in two main components:

- A configuration Tool
- The Scheduler program

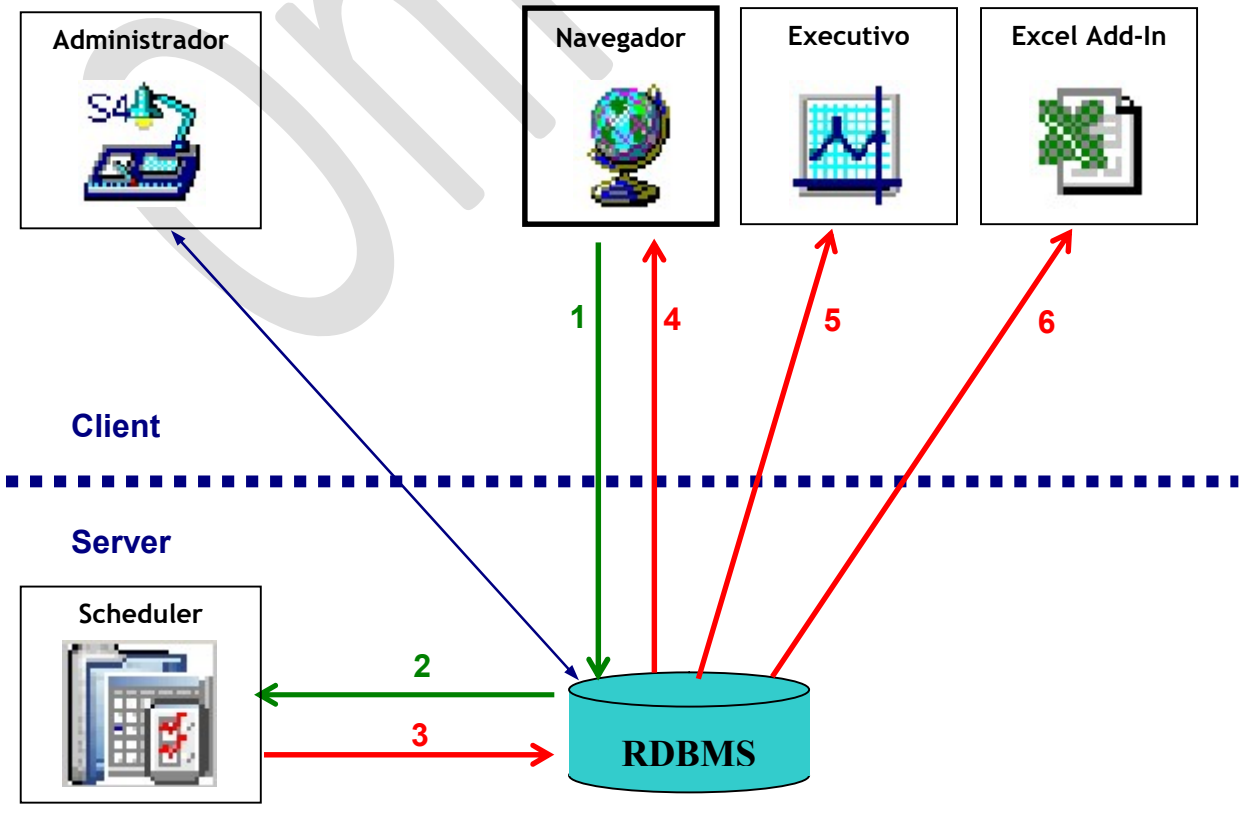
The Configuration tool is used to define and test the RDBMS connection settings and to define the logging level of the Scheduler.

After the settings are correctly defined, it is possible to launch the Scheduler successfully.

The jobs' window execution, the scheduler pooling and other administrative tasks are defined in the BIS-Administrator.

In the picture bellow it is visible that the Scheduler runs as a server component, like the database server, while the other BIS tools run in Windows Workstations.

The Scheduler can be installed in the same machine where the database server is running or in a separate machine. The scheduler was implemented as a Java application. This allows it to run in any platform where a Sun ™ Java Runtime Environment 1.4.2 or above is supported. The communication between the Scheduler and the database server is established using a JDBC driver, and the RDBMS specific connectivity layer, both available from the RDBMS provider.
As an example, for Informix Dynamic Server, you must use the free Informix Connect middleware and free Informix JDBC driver provided by IBM.



Before the users schedule their tasks, the Administrator can do a set-up specific for the company. The Administrator, can define:

1) Which are the weekdays and the periods in those days when it is allowed to launch scheduled jobs?
2) The pooling interval of the Scheduler. The time that the scheduler sleeps after every validation to find if there are jobs on time to be launched.
3) Which are the tables' cache cleanup policies.

A user can submit a task to be executed by the Scheduler even when the scheduler is not run.

The scheduler can be configured to start automatically, as soon as the machine starts, or manually. The BIS Administrator can manually notify the scheduler to stop, using the stop button of the Scheduler Administration Console in the Windows BIS-Administrator tool.
The Scheduler can be configured to stop when the system is shutting down.

The flow to get results generated by the scheduler for a submitted task by the BIS-Navigator is the following (see the numbered steps in the above picture):

1) The BIS-Navigator submits the Report task to the scheduler and defines which are the groups or individual users that will have access to the generated report. Several users can use this way one single report execution. The submitted task request is registered in the s4_crontable table.

2) The S4-Scheduler, periodically (the period is defined by the pooling interval), verifies if there are tasks that match the time criteria to be launched.

   The S4-Scheduler is based on a multi-thread architecture, so a single Java process can launch and run several reports in parallel.

3) For every launched task it is created a record in the s4_cron_jobs table. This table contains status and historic information about the executed tasks and still in execution. The execution of the task creates a cache table, with the results of the scheduled report. After the task is finished, the users with permissions to access to the results of this task are notified: A message box is automatically displayed in the Windows machine if the user is using one of the BIS client tools at that moment. If the user was not connected, he will be notified, in the next time he connects to one of the BIS client tools.

4) After being notified the user can read the results directly from the cache table. In our example the **step 4** shows a BIS-Navigator reading the results cached table; **step 5** represents another user using BIS-Executive reading the same cached table and **step 6** shows another user connected to BIS-for-Excel also using the same table.

Long running, complex queries can, this way, be scheduled to run in non-peak hours and be ready to be consumed by several users in a more efficient way.
There is also the possibility to define a specific Database TableSpace to store the cache tables, what can minimize the database I/O concurrency with the regular OLTP activity.

Another feature is the possibility to generate XML formatted output files. These XML files will be stored in the Server Machine folder/directory defined by the BIS-Scheduler Set-up tool. The Filename is generated using the following naming convention:
<Cube Name>_<Author LoginId>_tid#<Task Id Number>.xml

As an example a file with the name e1_n1_tid#51.xml was defined by the following constants:

1) Cube = e1
2) User Login = n1
3) Task Id = 51

Installation requirements

S4scheduler server tools were developed with **JAVA technology**.

The **S4Scheduler** was tested using Sun Java Runtime Environment JRE 1.4.2 and 1.5.0_06 with the Type 4 JDBC drivers provided in the BIS Software CD.

The tests were conducted on Windows 2000; Windows XP and Mandrake Linux, but the **S4Scheduler** can run on any platform where the **Sun Java Runtime Environment** JRE **1.4.2** and **1.5** are supported.

The **S4scheduler** server tools are distributed in an **s4scheduler.tar** file located under the **s4Scheduler folder** of the BIS **Software CD**.

The same file can be uncompressed on both Unix or Windows platforms.
For **Linux/Unix** use **tar –xvf s4scheduler.tar** and WinZip or WinRAR in **Windows.**

The s4scheduler.tar file contains the following folders/files:

S4SchedulerApp/init.d:	Contains a Shell Script for the automatic Start/Stop of S4Scheduler on Unix/Linux.
S4SchedulerApp:	The s4Scheduler application.
S4SchedulerApp/lib	The JDBC Drivers for all supported databases.

NOTE: You **MUST** install the **SUN JRE 1.4 or 1.5** appropriate for your OS and Database before installing the S4IT Scheduler.

NOTE: There is **NO** need to install any Database Connectivity Software, because all the **JDBC Drivers** provided with S4IT Scheduler are **Type 4**. This means that they already include the embebbed RDBMS connectivity.

Setup on Unix/Linux

This section describes how to install the S4scheduler in a Unix/Linux server.

S4scheduler Installation

To install the S4scheduler execute the following steps:

1. Login as super user.
2. Copy the s4scheduler.tar file to a temporary folder (/tmp).
3. Create the destination folder. In our example is /opt/S4.

```
mkdir  /opt/S4
```

4. Extract the contents of the s4scheduler.tar file to /opt/S4.

```
cd   /opt/S4
tar   xvf   /tmp/s4scheduler.tar
```

5. Edit the S4setup.ini file.

```
cd   S4SchedulerApp
vi   S4setup.ini
```

6. In **S4setup.ini** file replace **S4** by the folder **(/opt/S4/setup)** .

   Edit the ***.sh** files and define the **S4IT_SCHEDPATH** and **PATH** environment variables.

   The **S4IT_SCHEDPATH**: defines where the scheduler **java_exec files** are located

   The **PATH:** must include the bin folder PATH for the SUN JRE in use

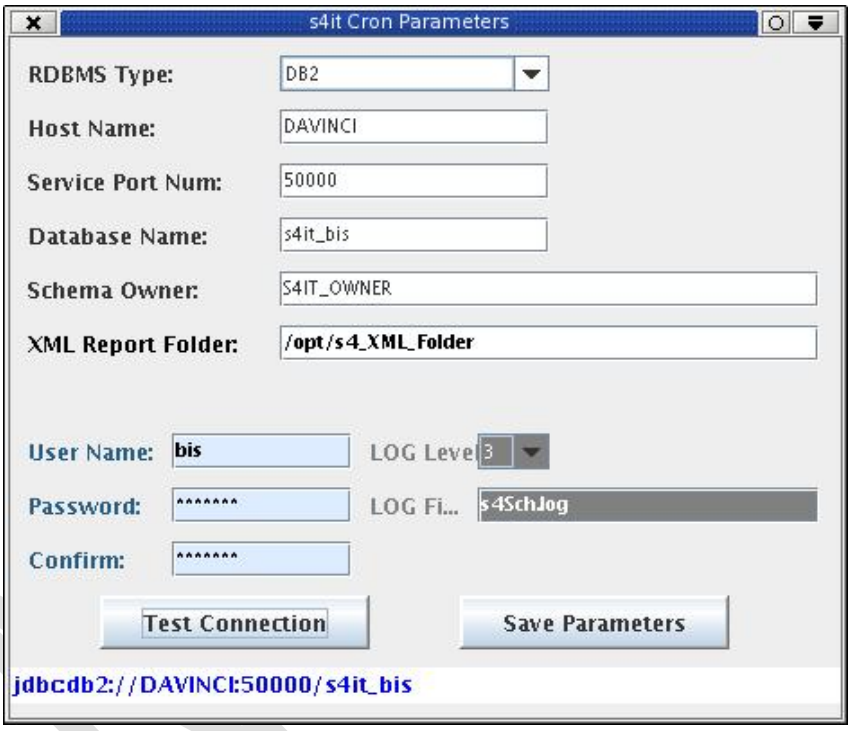
Configure the S4Scheduler Properties

1. Launch the S4 Scheduler Parameters Set-up Tool.

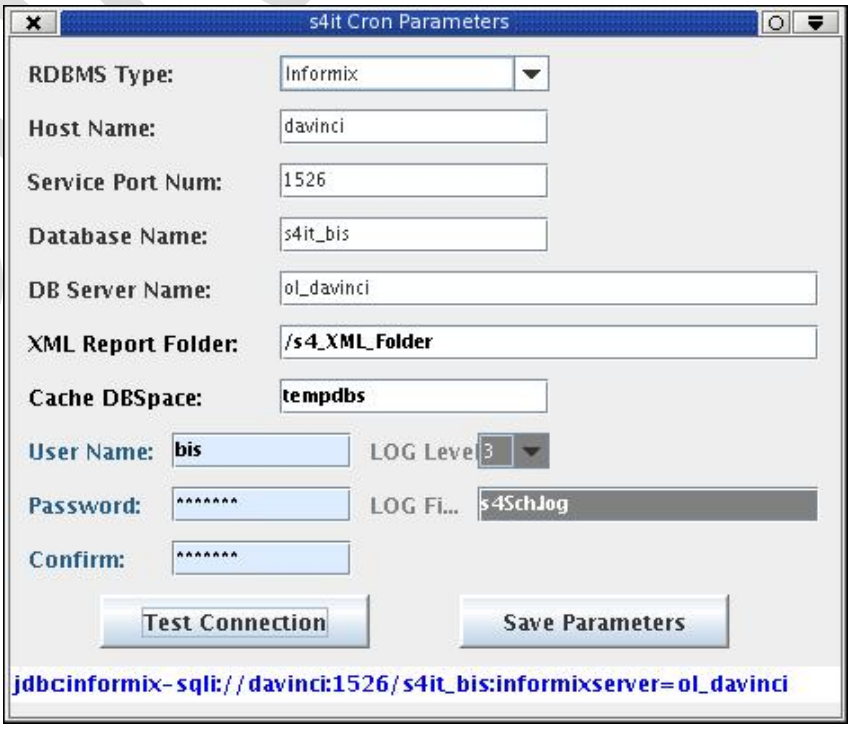
   cd /opt/S4/S4SchedulerApp
   ./s4CronParams.sh

   A form with fields that depend on the database type is presented.

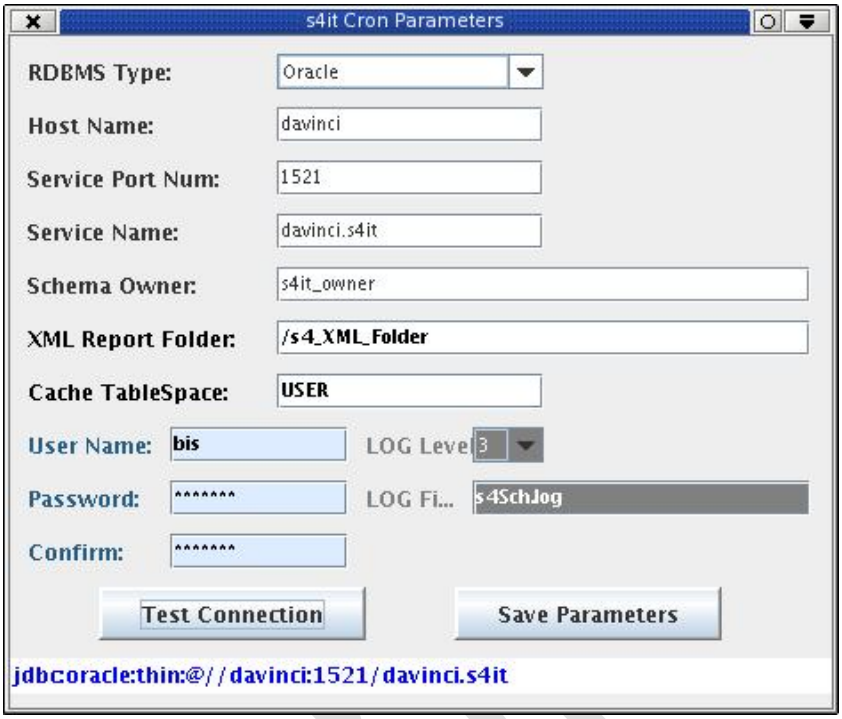
IBM DB2 Sample configuration



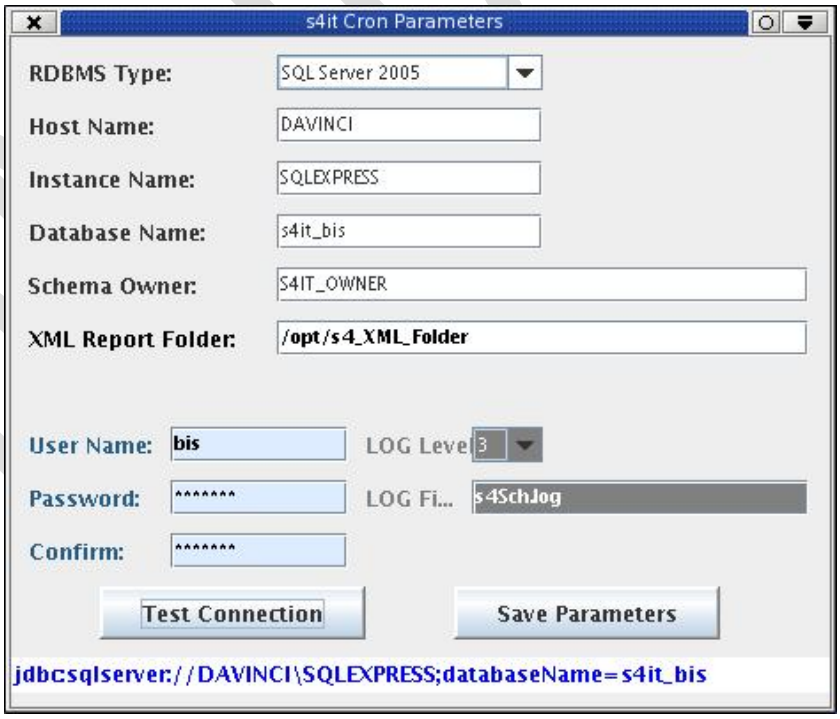
IBM Informix IDS Sample configuration



Oracle Sample configuration



SQLServer 2005 Sample configuration



The JDBC Drivers and URL connections for SQL Server 2000 and 2005 are distinct, so choose the correct RDBMS Type in the list before complete the setup. If you configure the connection for SQLServer 2000 and you attempt to connect to 2005, you will get the following error:

In this form is necessary to define correctly the connectivity parameters for your specific database. For a DBA all these parameters are well known, so this configuration is usually an easy task.

A simple method to validate if the Database is listenning in the expected port is to run a TELNET command:

TELNET <RDBMS Server IP or Hostname> <DB Listenning PORT number>

Recommendation: In the first set-up set the LOG Level to 3. This will help you to get more information if you get into some errors. After you have finished testing the S4Scheduler, then, change the Log Level back to zero or one.

Initially you can press the button Save Parameters and then Test Connection.

If the connection succeeds, then you will get the following message box:

and then

After a successful test the Properties are automatically saved.

Some typical errors are:

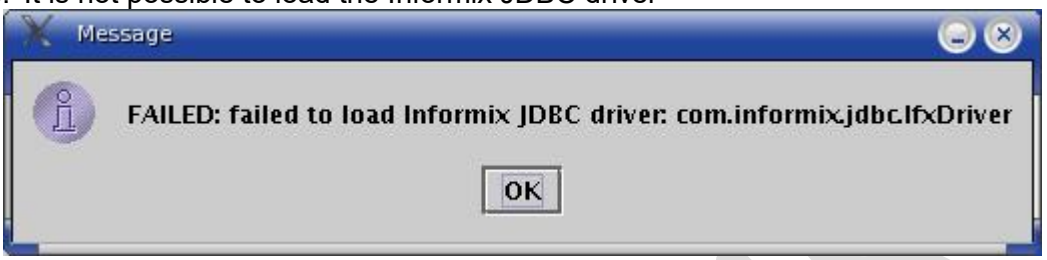
1. The database is down

2. Username/Password errors

3. Probably the database name is not correct or the user doesn't have access permissions to the database

4. The RDBMS Specific field (INFORMIXSERVER) is not correct

5. It is not possible to load the Informix JDBC driver



If the connection test was successfully done then we can test the launch of the S4Scheduler

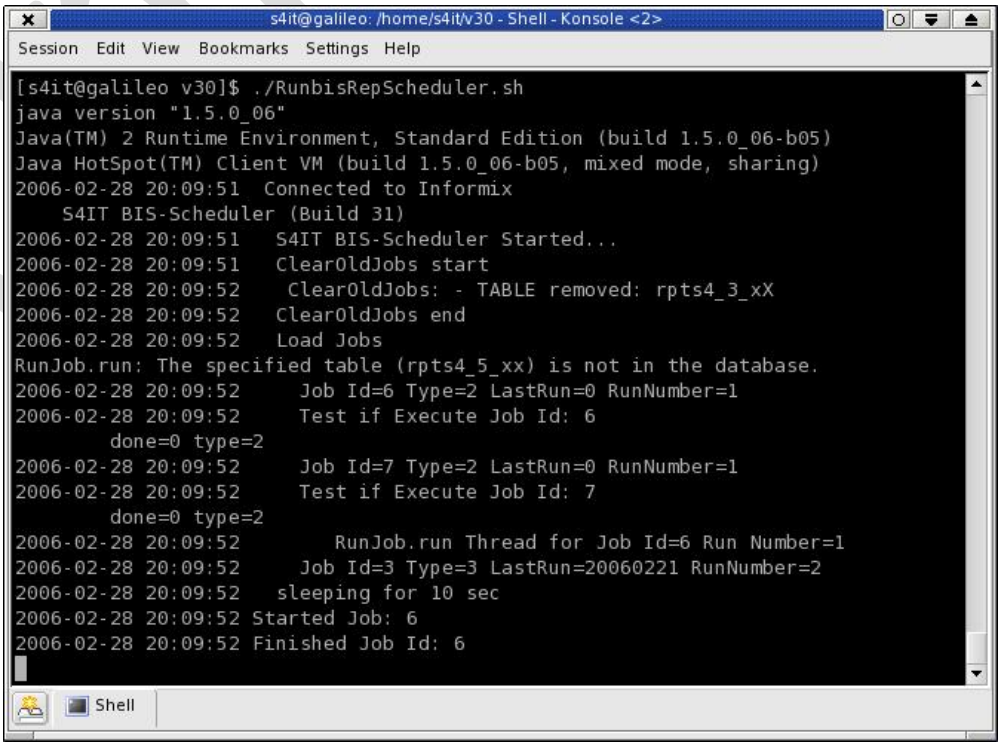
Launch S4Scheduler

You can launch the S4Scheduler in two ways: automatically during the UNIX start-up or manually.

Manually start-up

```
$ cd /opt/S4/S4SchedulerApp
$ ./RunbisRepScheduler.sh
```

The S4Scheduler with a Log Level of 3 should show an output similar to the following:



With Log Level 0 (zero) you will get only the following output:

2006-02-20 11:53:31 S4IT BIS-Scheduler Started...

Automatic start-up

To perform an automatic S4Scheduler start-up and stop you must copy the inid.d/s4scheduler file to your UNIX **system init.d** folder and then make the link to the correct **rcX.d** level files. The **s4scheduler** file must have at least "**rwx------"** permissions to the **root user**.

Consult your Linux/Unix Administrator for details on how to set-up.

As an example in a Linux system where the default initstate (startup level) is 5 and the shutdown level is 6, you should create the following links:

ln -s /etc/init.d/s4scheduler /etc/rc5.d/S99s4scheduler
ln -s /etc/init.d/s4scheduler /etc/rc6.d/K00s4scheduler

This will allow the system to start the S4Scheduler after all other services are running and to stop it before the other services are stopped.

Setup on Windows

This section describes how to install S4scheduler on a Windows server.

S4scheduler Installation

To install S4scheduler execute the following steps:

7. **Login** as super user.

8. **Copy** the s4scheduler.tar file to a **temporary folder** (c:\temp).

9. Create the destination folder. In our example is c:\S4.

   cd c:
   mkdir S4

   Using WinZip or WinRAR **extract** the contents of the s4scheduler.tar file to c:\S4.

   Edit the ***.bat** files and define the **S4IT_SCHEDPATH** and **PATH** environment variables.

   S4IT_SCHEDPATH: defines where the scheduler **java_exec files** are located
   PATH: must include the **bin** folder PATH for the **SUN JRE** in use

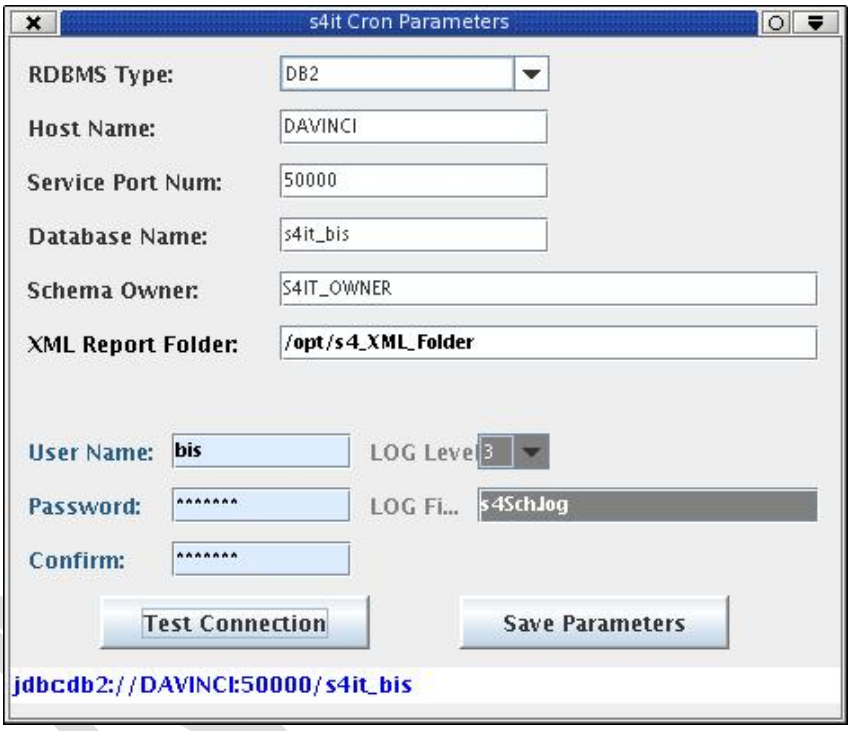
Configure the S4Scheduler Properties

2. Launch the S4 Scheduler Parameters Set-up Tool.

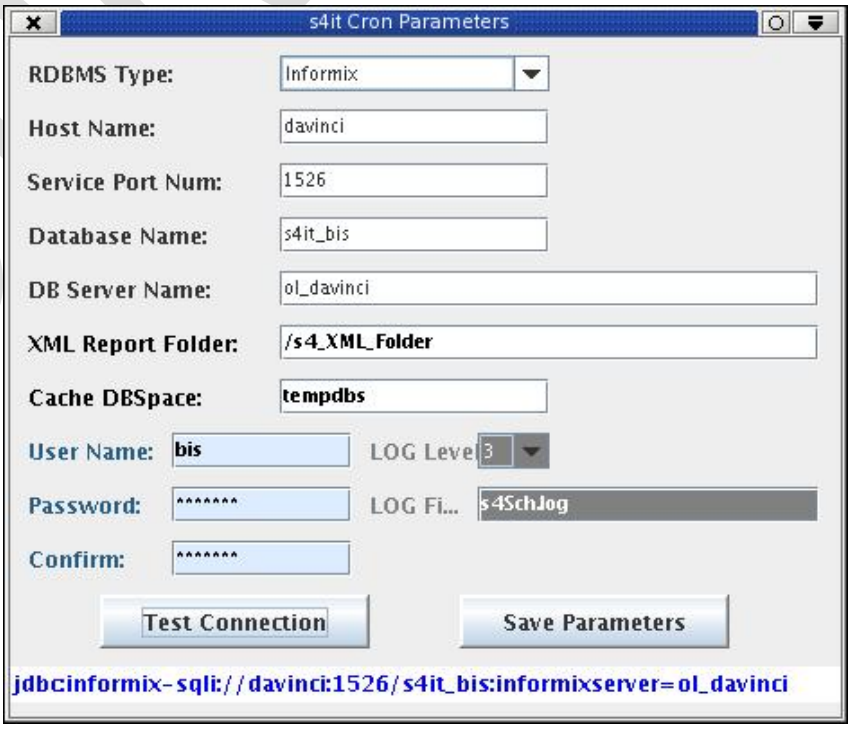
```
cd   /opt/S4/S4SchedulerApp
./s4CronParams.sh
```

A form with fields that depend on the database type is presented.

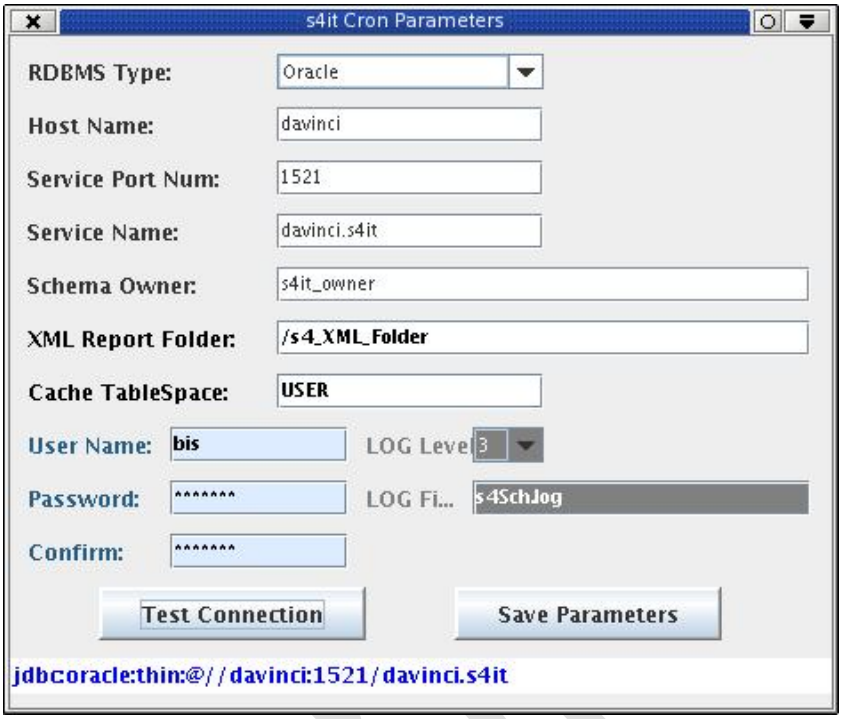
IBM DB2 Sample configuration



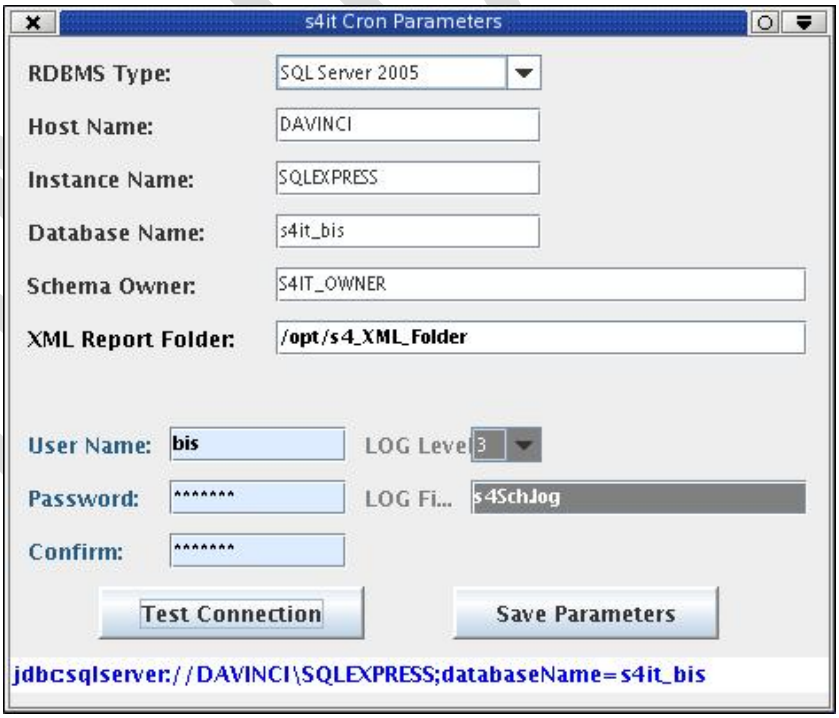
IBM Informix IDS Sample configuration



Oracle Sample configuration



SQLServer 2005 Sample configuration



The JDBC Drivers and URL connections for SQL Server 2000 and 2005 are distinct, so choose the correct RDBMS Type in the list before complete the setup. If you configure the connection for SQLServer 2000 and you attempt to connect to 2005, you will get the following error:

In this form is necessary to define correctly the connectivity parameters for your specific database. For a DBA all these parameters are well known, so this configuration is usually an easy task.

A simple method to validate if the Database is listenning in the expected port is to run a TELNET command:

TELNET <RDBMS Server IP or Hostname> <DB Listenning PORT number>

Recommendation: In the first set-up set the LOG Level to 3. This will help you to get more information if you get into some errors. After you have finished testing the S4Scheduler, then, change the Log Level back to zero or one.

Initially you can press the button Save Parameters and then Test Connection.

If the connection succeeds, then you will get the following message box:

and then

After a successful test the Properties are automatically saved.

Some typical errors are:

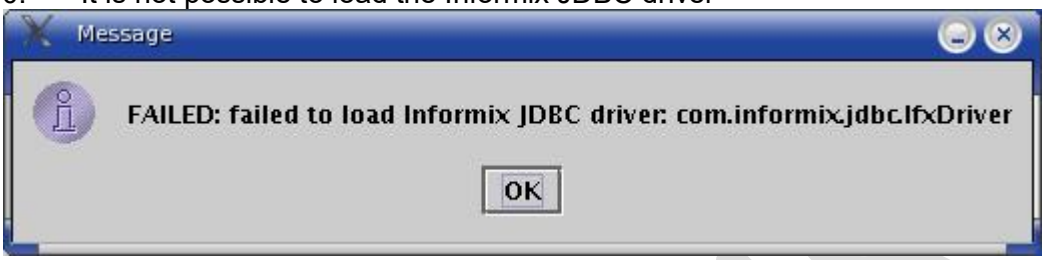
6. The database is down

7. Username/Password errors

8. Probably the database name is not correct or the user doesn't have access permissions to the database

9. The RDBMS Specific field (INFORMIXSERVER) is not correct

10. It is not possible to load the Informix JDBC driver



If the connection test was successfully done then we can test the launch of the S4Scheduler

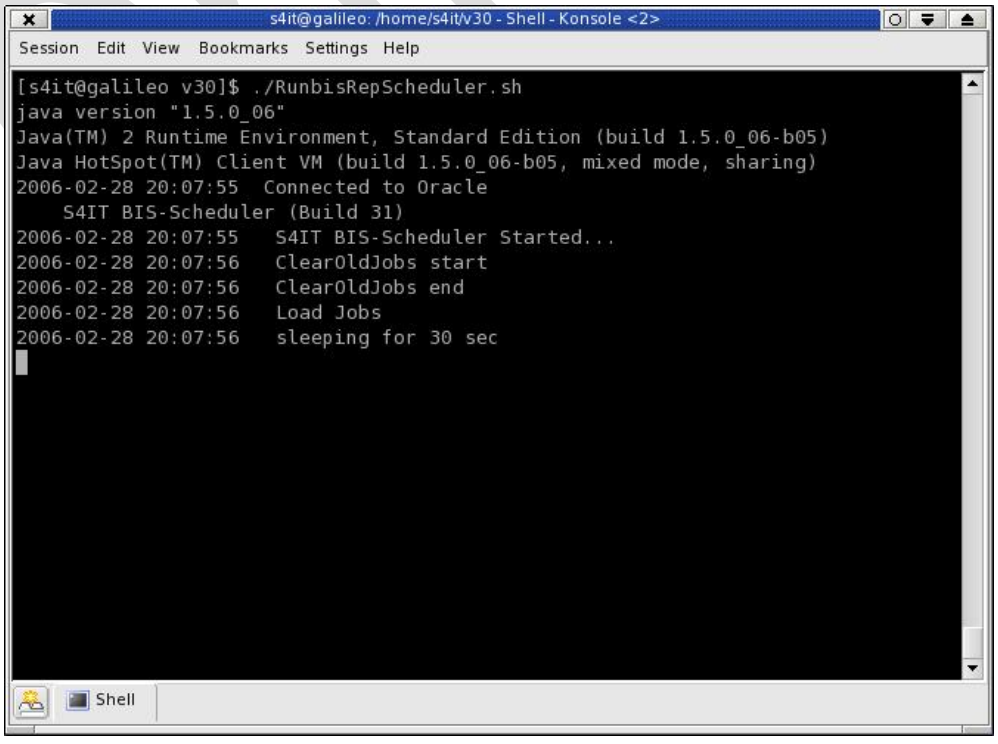
Launch S4Scheduler

You can launch the S4Scheduler in two ways: automatically during Windows start-up or manually.

Manually startup

cd c:\S4\S4SchedulerApp
run RunbisRepScheduler.bat

The S4Scheduler with a Log Level of 3 should show an output similar to the following:



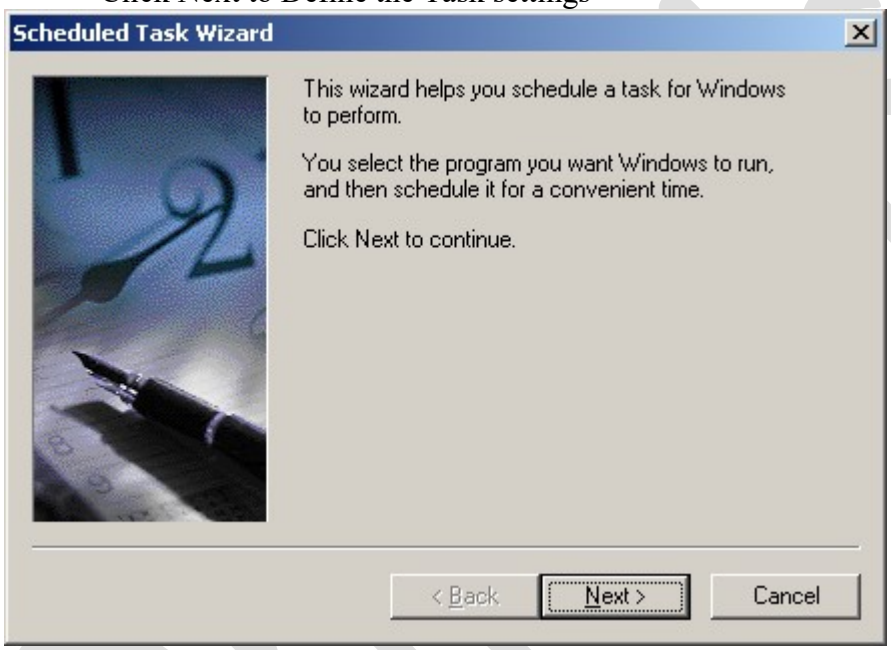
With Log Level 0 (zero) you will get only the following output:

2005-07-03 11:53:31 S4IT BIS-Scheduler Started...

Automatic startup

To perform an automatic S4Scheduler when windows server starts do the following as Administrator.

- In the Control Panel open the Scheduled Tasks.
- Then Add a Scheduled Task
- Click Next to Define the Task settings



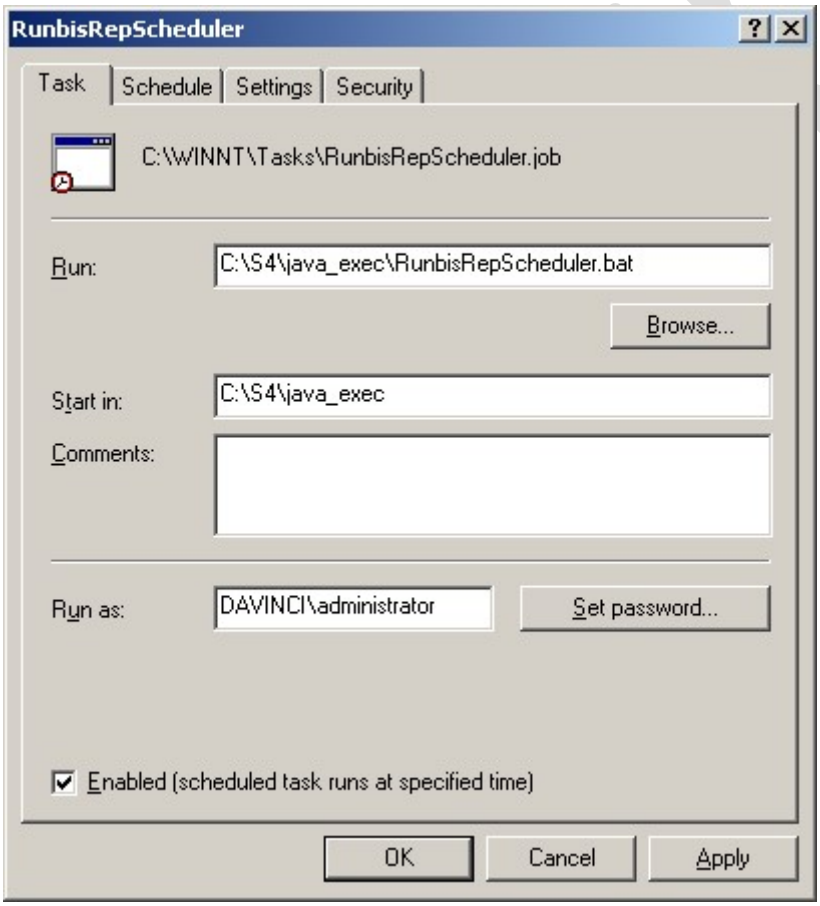
- Find the file **RunbisRepScheduler.bat**

- Choose to perform the task when the Computer starts

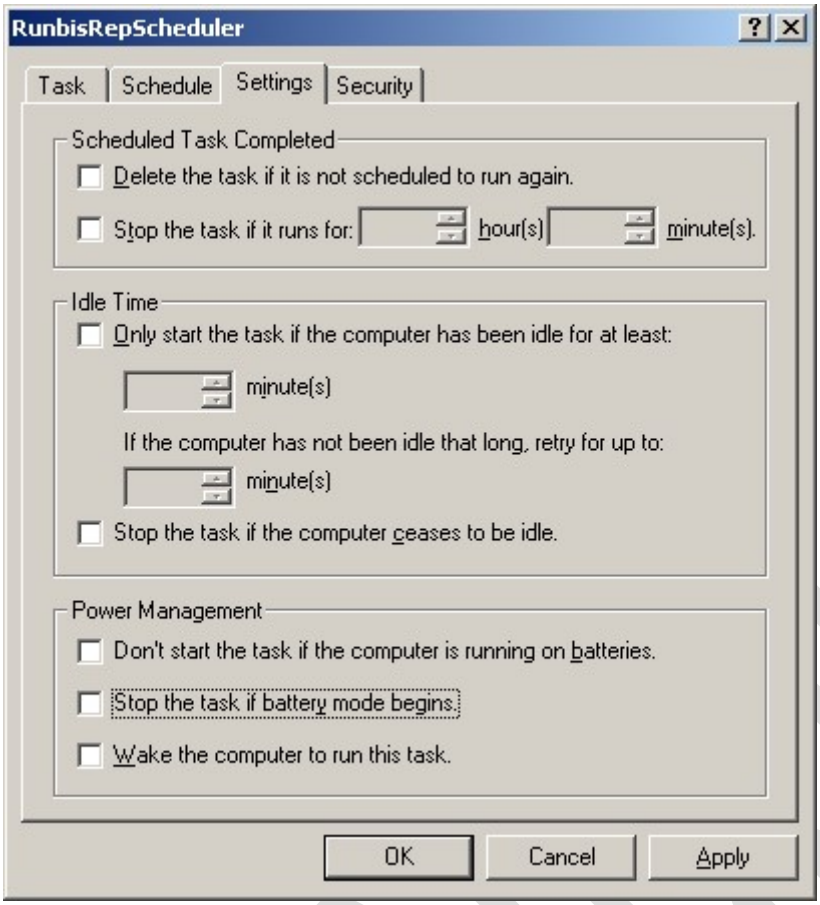
- Define a user with permissions to launch the task and write in the folder where the **RunbisRepScheduler.bat** is located**.**

- Check the **Open Advanced Properties** box**.**

- Check the **Open Advanced Properties** box and then click Finish**.**



- In the **Setting Tab** **uncheck** all the boxes and then **apply** the changes**.**


RunbisRepScheduler
Task
Schedule
Settings
Security
Scheduled Task Completed
Delete the task if it is not scheduled to run again.
Stop the task if it runs for:
hour(s)
minute(s).
Idle Time
Only start the task if the computer has been idle for at least:
minute(s)
If the computer has not been idle that long, retry for up to:
minute(s)
Stop the task if the computer ceases to be idle.
Power Management
Don't start the task if the computer is running on batteries.
Stop the task if battery mode begins.
Wake the computer to run this task.
OK
Cancel
Apply