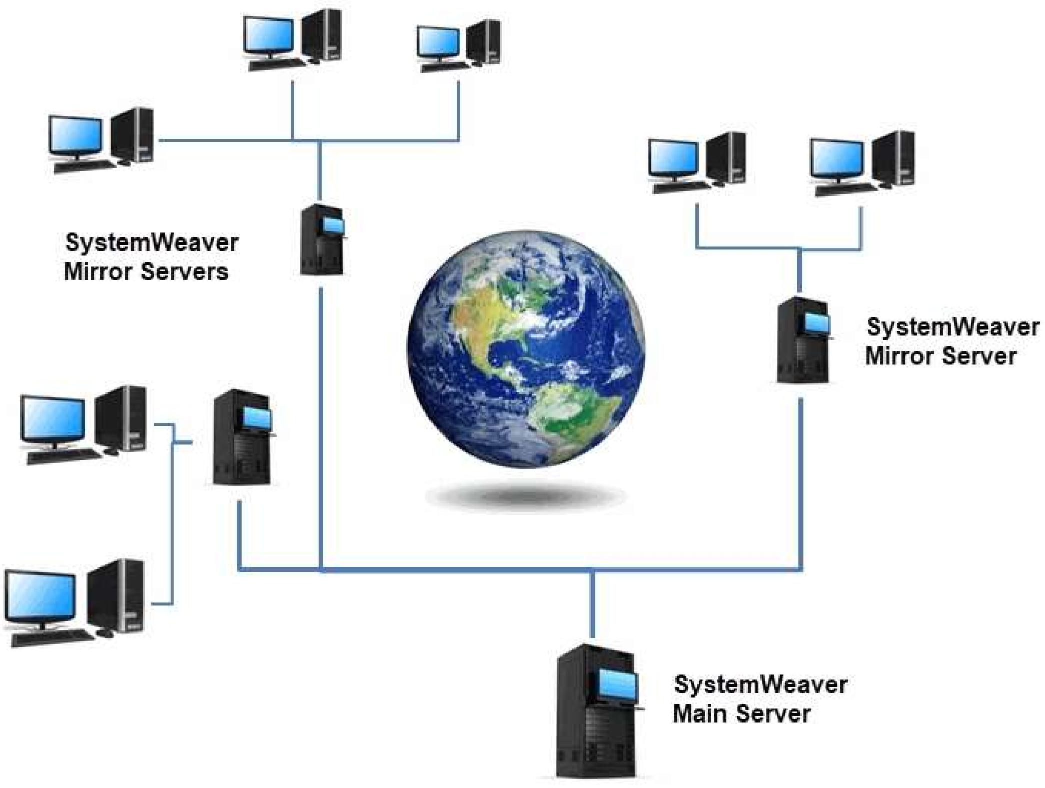
SystemWeaver®

Version: R44
Classification: Client Confidential
Date: 2023-10-13

SystemWeaver Administration Manual



SystemWeaver ID: x04000000000D2CB3

1 General Information

This User Manual covers administration procedures in SystemWeaver® corresponding to SystemWeaver version 1.53 and later. The manual is supplied as part of a SystemWeaver License Agreement and may only be copied and redistributed in accordance with the License Agreement.

1.1 Introduction

This instruction covers all required common activities and actions performed by a SystemWeaver Administrator.

SystemWeaver databases usually represent a considerable value to the user organization, and a continuous fault free operation may be critical to the organization. Anyone who administers a SystemWeaver installation base should have received the proper training in SystemWeaver administration. Refer to Systemite AB or the SystemWeaver Application Manager at your company for SystemWeaver Administrator training.

Note that this instruction does not substitute proper training in SystemWeaver Administration, and should be used as a reference only.

This instruction supports SystemWeaver applications of version 1.51 and later.

Systemite AB reserves the right to change this instruction or the SystemWeaver software at any time, without notice, and the manual may differ from the current tool version.

1.2 System Requirements

1.2.1 General

Software Requirements

SystemWeaver servers can be installed and operated on any 64 bit Windows server or PC (Ancient versions, 7, 10), i.e., no server edition of Windows is required, although any Windows server version works as well.

In SystemWeaver, most of the computation is performed by the clients, meaning that the computational requirements on the server is low. In fact, any recent PC could work as a SystemWeaver server host even for large projects, provided the other system requirements are satisfied.

If not already installed, the server requires .NET Framework version 4.7 or later.

Visual C++ Redistributable for Visual Studio is often already installed on Windows machines because many other programs also use it. But if a machine is basically clean with not much installed, it may be missing. If that is the case, you will need to install it. Version 2017 or later is needed.

CPU

The SystemWeaver server can run on any CPU, but needs 2 cores to function. At least 8 is recommended.

What is important for maximal performance is single core performance. So if optimizing, go for CPUs with high single core performance rather than CPUs with many cores.

Note: You will not necessarily get better performance on hardware that has, for example 256GB memory, 16 core (3GHz for each core) and I TB SSD, compared to a “normal” server (64GB memory, 8 core and ITB HDD). A faster CPU is a determining factor. More cores or more memory are not the determining factors, unless you reach some ceiling, e.g., very large number of connections or swapping memory.

Port
With regards to port requirements, SystemWeaver only needs the port that will be used to be open. For example, if users will be logging in to the main server using port 1234, then only 1234 needs to be open. If a user is logging in to a mirror server using port 5678 (which connects to the main server using port 1234, then only port 5678 needs to be open on the mirror server. Closing port 80 has no effect on SystemWeaver. We recommend chosing a number greater than 1000.

Location
The closer the main server is located to the users, the lower the network latency. If the main server must be in a different geographic location, or if users are spread out geographically, we recommend implementing our mirror server solution.

1.2.2 Database

Persistent storage of the SystemWeaver database is done in an SQL database.
The following SQL databases are currently supported:

- SQLite
- Oracle
- SQL Server

Regarding questions about specific versions of these, contact Systemite.

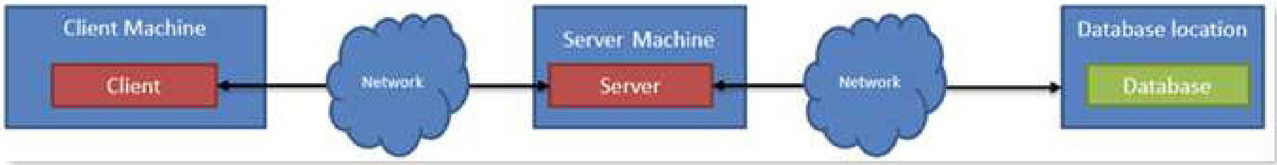
1.2.2.1 Checklist for MS SQL Database

This section provides some configuration support for implementation and maintenance of a MS SQL database for SystemWeaver.

Ensure Stable Connection
To avoid unwanted consequences (e.g.,corrupt data) caused by the SystemWeaver server attempting to continue to run during network outages, etc., the server will shut down and must be started again if the environment is unstable. To avoid such restarts, it is of extreme importance that the connection between the SystemWeaver server and the MS SQL database server be kept stable.

Tip: To guarantee a stable connection, consider using an SQ Lite database.

Low Latency
When clients perform Write operations, the SystemWeaver server must connect to the MS SQL server Because it is external, i.e., resides on a different server than the SystemWeaver server application, there will be latency. Keep this in mind when determining the location of the MS SQL server in relation to the SystemWeaver server.



Having the SQL server on the same machine or close by the SystemWeaver server application will improve the latency. The best write performance is with an SQLite database on local SSD disk.

Max Text Replication Size
The max text repl size option in MS SQL specifies the maximum size (in bytes) of text, ntext, varchar(max), nvarchar(max), varbinary(max), xml, and image data that can be added to a replicated column or captured column in a single INSERT, UPDATE, WRITETEXT, or UPDATETEXT statement. The default configuration for this setting in MS SQL is 65536. Consider increasing this size if users will be adding data in excess of that.

Database Authentication
The DatabaseUser entered in the server configuration must have sufficient authorization in the SQL database. You can use a service account in Active Directory for MS SQL server authentication.

ODBC Driver
To avoid server stops due to attempts to save too large of an amount of data to a text or image column in MS SQL, be sure to upgrade to a newer version of Microsoft ODBC Driver for SQL Server, e.g., Microsoft® ODBC Driver 13 for SQL Server® - or later. There is a Check max blob size tool in the swDatabaseManager utility to test importing large data. To test, click the button. You do not need sole access to the database. The ODBC driver is installed on the machine where the SystemWeaver server application is installed.

1.2.3 Storage and Memory

SystemWeaver uses a proprietary, high performance, object oriented, in-memory database that uses virtual memory for storing the entire database. This means that the size of the virtual memory limits the possible size of the database. Since Windows does not handle excessive memory-use very gently, this limit should be monitored regularly, e.g. by using monitoring functions in Windows like the Task Manager, or the option described in SystemWeaver Server.

SystemWeaver is very efficient in its data handling (efficient transformation of information into data) and storage so a Gbyte will go a long way. As a rule of thumb, a full time systems developer, who maintains all development data in SystemWeaver, produces an average of 200 kbyte of data per month*, even for demanding use cases. The linear characteristic of memory consumption per time unit means that Moore's law (and Kryder's law) assures that time is on our side in this case.

Note that the actual use of memory can be observed in the log file of a SystemWeaver server.

The 64 bit SystemWeaver server applications are in practice limited only by the amount of available RAM, since the address space is virtually unlimited.

The access performance is affected by the available RAM. If too little RAM is available, parts of the database may become swapped out to disk which will drastically reduce performance. Considering the low cost of RAM however, at least 8 GB of RAM should be available to the SystemWeaver server applications, however, the size of the database is also a decisive factor. The memory allocated to the main server should be at least 3 times the size of the database. For example, if the database is 10 GB, you need at least 30 GB RAM for the server to function. If slave servers are included in the installation, the memory allocated to the main server should be equal to the number calculated as described above (e.g., 30 GB) x 3. So, for this example, 90 GB. It is recommended that regular monitoring of the resource utilization is performed in order to detect the need for upgrades of the server platform.

Installation Disk Size
The actual software needs 100MB of disk space.

Data Disk Size
This depends entirely on your use case. How much data will you be working with and storing in the system? We have installations from 100MB up to 10s of GB. In addition to the actual memory for storing the database, you should make sure that you have the means to backup the database somewhere. If you are using a SQLite database, the data disk size applies to the SystemWeaver application server since the database is ideally stored there (for optimized performance) and not on a different server machine.

File Repository Disk Space
If you want to use the file repository feature, i.e., the ability to add attachments to items and issues, you'll need disk space for this. The attachments are not stored in the main database but rather in a separate, single sqlite database or in a dedicated file structure (the later is recommended). Regardless of which option is selected, the repository must reside on local disk on the same server machine that is running the SystemWeaver server executable. The size needed depends entirely on how many and how large of files your users will put into the system. This can be everything from 10MB to terabytes. We recommend 300 GB of free disc space. If you predict that this will be an often-used feature, we recommend that you put this on a partition were you can increase as the needs of the users increase.

* According to statistics from reference user organizations during a 10 year period. More precise estimates need to be based on specific use cases.

1.2.4 Virtualized Servers

SystemWeaver server application can be run in virtualized server environments like VMware (of VMware Inc.). However, such environments may not be intended or optimized for in-memory databases, and as a result performance may be less-than optimal*.

It is recommended that you consult Systemite before deploying production environments in virtualized environments.

* It has been observed that virtualized environments do not always respect reserved RAM quota, which may severely deteriorate performance.

1.2.5 DLLs

Some SystemWeaver applications may require specific DLLs. When such DLLs are not part of standard Windows configurations they are included in the distributions from Systemite.

There may be special cases when DLLs are installed to a majority of Windows computers, and therefore not included in the distributions. Read more about such cases for each specific application in the section titled SystemWeaver applications and utilities.

1.2.6 Recommend server alias

We recommend the use of a server alias not only to hide the path to your data, but also to allow for smoother machine changes for your SystemWeaver server installation. A server alias is especially suitable for:

- Use of a ClickOnce Client wrapper
- Use of the swExplorerLauncher (see SystemWeaver URL)

- Integrations with other tools where you store the server name
- Easy distribution of server address to users

An alternative to aliases is, of course, to register a domain name (a record), which is the method you must use for accessing the system outside your local network/VPN.

2 Server Configuration and Installation

SystemWeaver includes a number of server applications that can be configured and used in many different ways, depending on the needs of the user organization and IT support organization.

2.1 Overview of SystemWeaver Applications

This paragraph gives an overview of the various SystemWeaver applications. The following paragraphs describe these applications in more details.

The following SystemWeaver applications are available in a 64 bit version:

swDBServer	Server	A SystemWeaver Application Server manages all SystemWeaver data and offers connections from any SystemWeaver application through the SystemWeaver API (The swDBServer runs as a background process in Windows, with interfaces to server configuration files, the database, a log file, and a status window. A SystemWeaver Server assures the integrity and security of the stored data, regardless of the operations performed by connected Clients.)*. The application is used both as a mirror server as well.
swTestServer	Server	offers the same services as the swDBServer, and in addition it has a GUI that offers manual selection of configuration options, possibility to see all logged in users, and a log window corresponding to the optional log file
swExplorer	Client	is the standard SystemWeaver client. This client can be configured in order to support a number of different development tasks, see the section about the swExplorer.
swArchitect	Client	used for modification and administration of SystemWeaver Meta Models. The swArchitect is described in more detail in the section about swArchitect.
swAdmin	Client	manages SystemWeaver user accounts and roles. The swAdmin is described in more detail in the section about swAdmin.
swNotificationServer	Client	is a SystemWeaver client** that automatically sends notification emails when the status has been changed of a SystemWeaver Issue (see ref [4]). The swNotificationServer is described in more detail in the section about SystemWeaver Notification Server.
swTestNotificationServer	Client	offers the same services as the swNotificationServer, and in addition it has a GUI that offers manual selection of server configuration options, a log window corresponding to the (optional) log file, and some special capabilities for testing purposes. The swTestNotificationServer is described in more detail in the section about the SystemWeaver Notification Server.
swDatabaseManager	Special	performs upgrades of SystemWeaver database schema. (The schema on this level is opaque to the user organization, and upgrades are only required when new SystemWeaver server releases requires such upgrades. Refer to the section about the swDatabaseManager for details.)
EncryptString	Utility	is used for the encryption of passwords. (The tool application has a self explaining GUI where any string is translated into an encrypted string. The encryption algorithm is based on 128 bit encryption.)
swServerMonitorService	Utility	application that supports multiple instances of the same SystemWeaver server as service on the same server computer

swServerCommand	Utility	command line utility that can send requests to a SystemWeaver server during runtime. Described in the section swServerCommand

In addition to the clients mentioned above, there are also many custom SystemWeaver clients. These clients use the same client-server connection and appear just like any standard SystemWeaver client. Note that these clients may install features, Item types and other meta data types on the server, provided that the user has the swArchitect rights.

* The SystemWeaver server consists of two executables - swDBServer.exe and Systemite.SystemWeaver.TcpSubServer.exe - with the first containing the connection to the database, cached data and all API logic, and the later managing communication through TCP with all the clients and their connections.
** Note that swNotificationServer is actually a client from a SystemWeaver topology aspect, although it acts as a kind of server towards email clients.

2.2 Versioning of SystemWeaver applications

The following parts of SystemWeaver have managed versions:

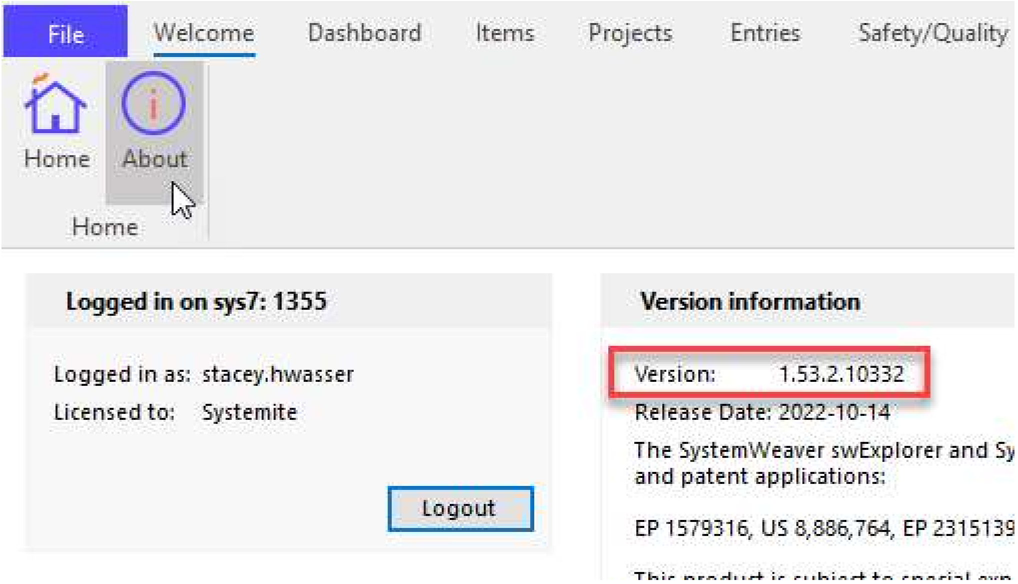
- Version of Application
- Version of client server API
- Version of Database

The compatibility of different versions in a SystemWeaver configuration is monitored and managed automatically by the SystemWeaver applications. In the case that the versions do not match, the client should detect this and inform the user of the problem and the required action.

2.2.1 Version of Application

Each application has a version that can be observed in the Welcome>About menu (for applications with a GUI) or in the Windows file Properties/Version, see examples below.

In the swExplorer Client:



Logged in on sys7: 1355

Logged in as: stacey.hwasser
Licensed to: Systemite

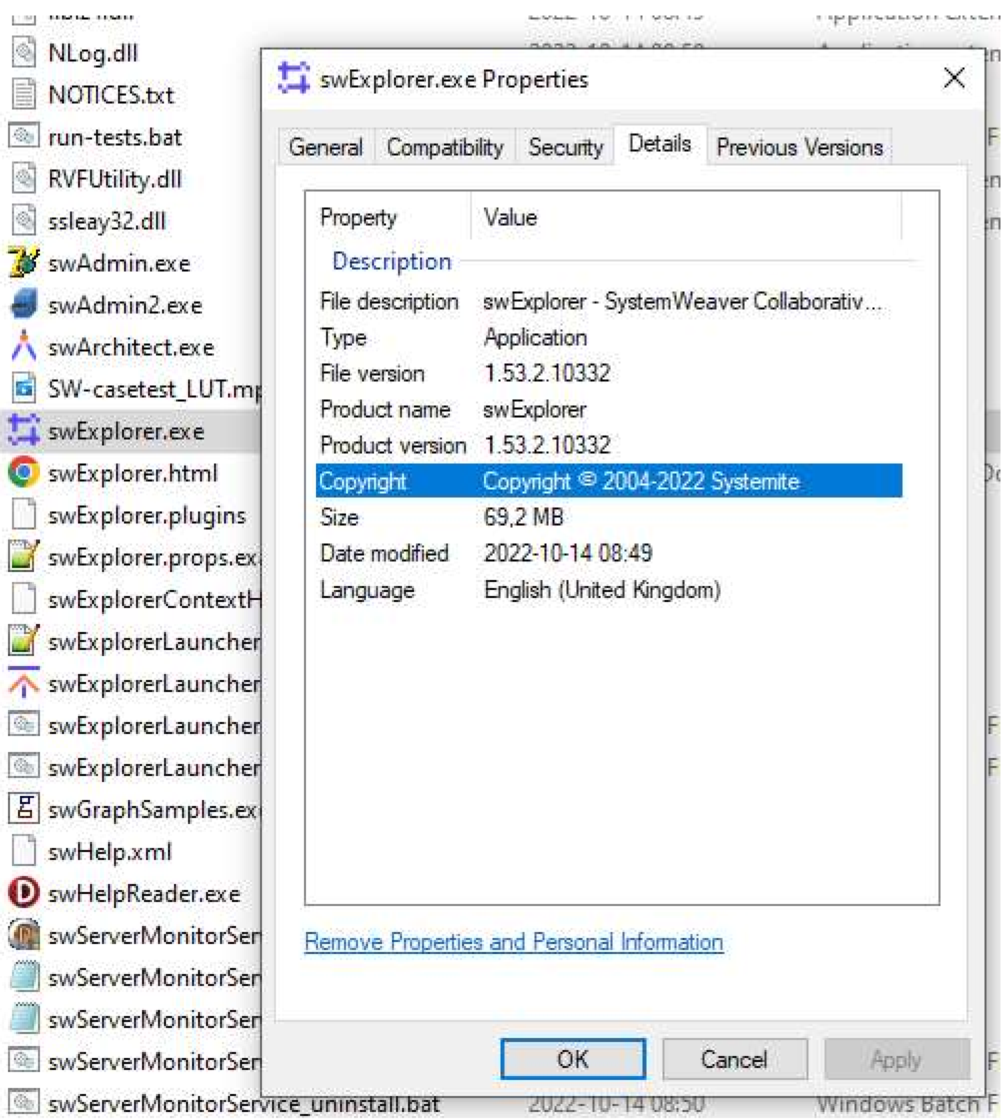
Version information

Version: 1.53.2.10332
Release Date: 2022-10-14
The SystemWeaver swExplorer and Sy
and patent applications:

EP 1579316, US 8,886,764, EP 2315139

Figure - SystemWeaver Version in swExplorer

As seen in Windows, file Properties:



Property	Value
Description	
File description	swExplorer - SystemWeaver Collaborativ...
Type	Application
File version	1.53.2.10332
Product name	swExplorer
Product version	1.53.2.10332
Copyright	Copyright © 2004-2022 Systemite
Size	69,2 MB
Date modified	2022-10-14 08:49
Language	English (United Kingdom)

Figure - SystemWeaver File Version

The version includes four numbers, separated by period:

- Main version: "1" in the example above
  Compatibility between applications requires identical main version
- Secondary version number: "52" in the example above
  Compatibility between applications requires identical main and secondary version
- Tertiary version number: "0" in the example above
  Compatibility between applications does not require the same tertiary version. For example 1.52.1.x is compatible with 1.52.2.x. However, it is recommended that the tertiary version numbers are identical.
- Build number: "4120" in the example
  The build number uniquely identifies the software build that produced the application.
  Build numbers do not have to be identical for the applications to be compatible, as described above.

2.2.2 Version of API

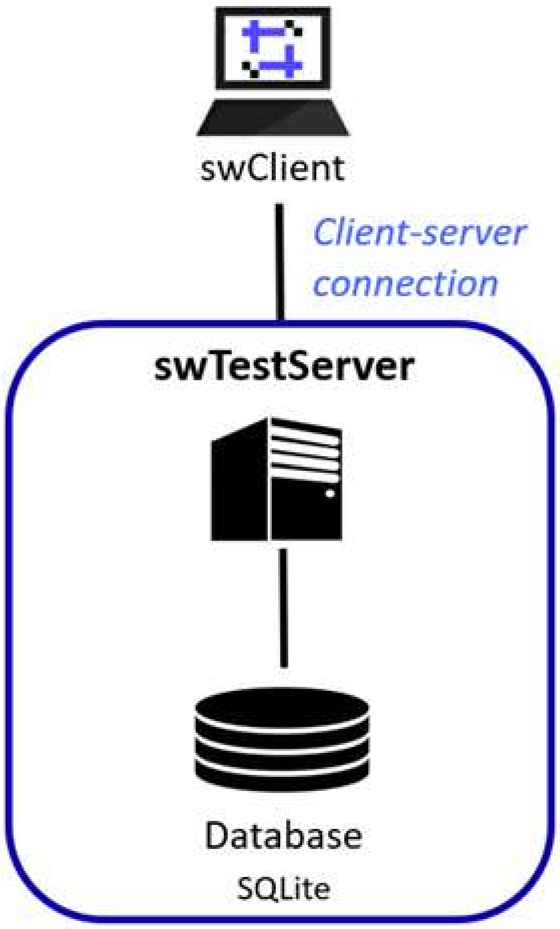
The API version identifies the version of the API between a SystemWeaver client and server. The API version is checked by the internal protocols between a client and server and is not visible to the end user.

2.2.3 Version of Database

The database version identifies the version of the internal database schema used by SystemWeaver. Each server version normally works with exactly one database version, except for the swDatabaseManager that is used for upgrading databases to current versions.

2.3 Alternative Configurations

The various SystemWeaver applications can be used in many different configurations, according to the needs of the organization. The figures in this paragraph illustrate some of these configurations.
The 'swClient' in the figures illustrates any kind of SystemWeaver client like a swExplorer, a swAdmin, or any custom SystemWeaver client.

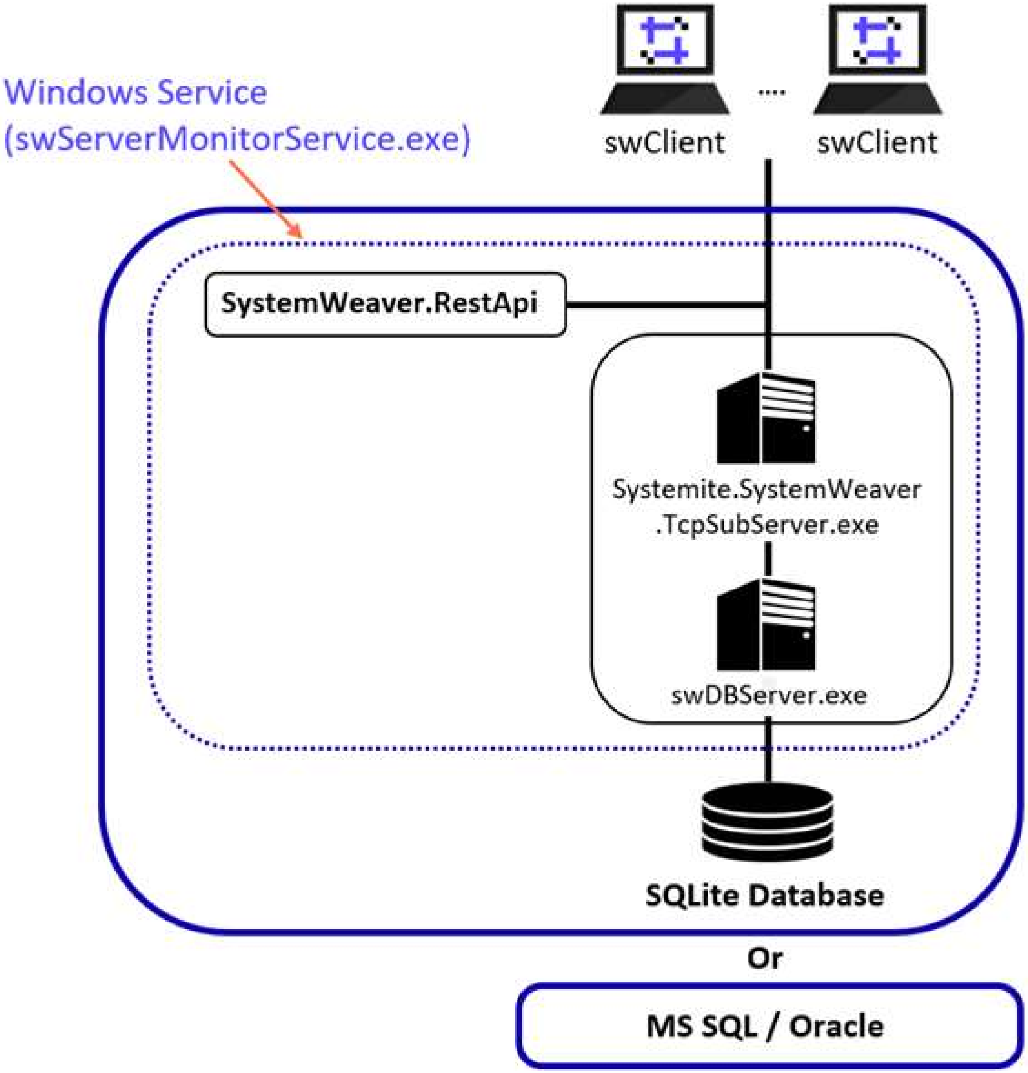


The TestServer configuration in the figure above illustrates the simplest SystemWeaver configuration.

The advantage of this configuration is that it is very easy to install on any PC using the GUI of the swTestServer in case any troubleshooting is needed. Typically any SystemWeaver installation will be tested first in this configuration, and later when the configuration has been tested successfully the configuration will migrate into any of the other configurations described below. The swTestServer is described in the section about the SystemWeaver Server.

2.3.1 Multiple Server Service Configuration

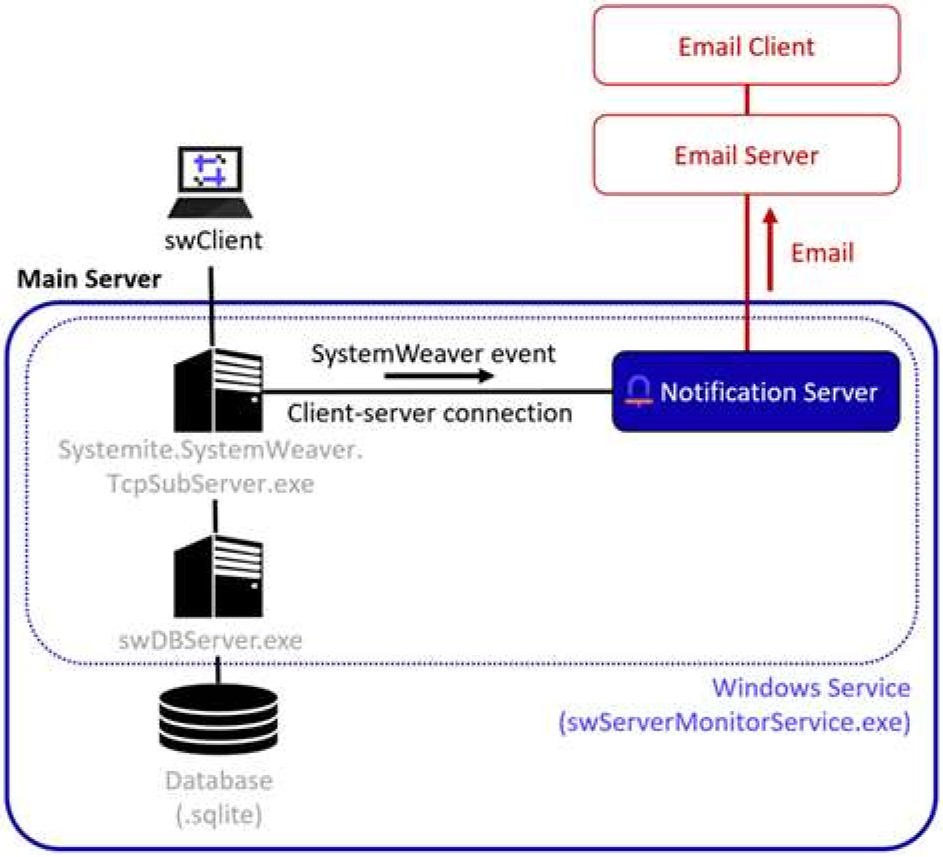
The swServerMonitorService monitors its server applications and can restart them when necessary. The swServerMonitorService supports single or multiple server instances. A use case for multiple instances would be, e.g., multiple test or QA installations, a server installation and a Rest API installation (as shown in the figure below), etc.



2.3.2 Notification Server Configuration

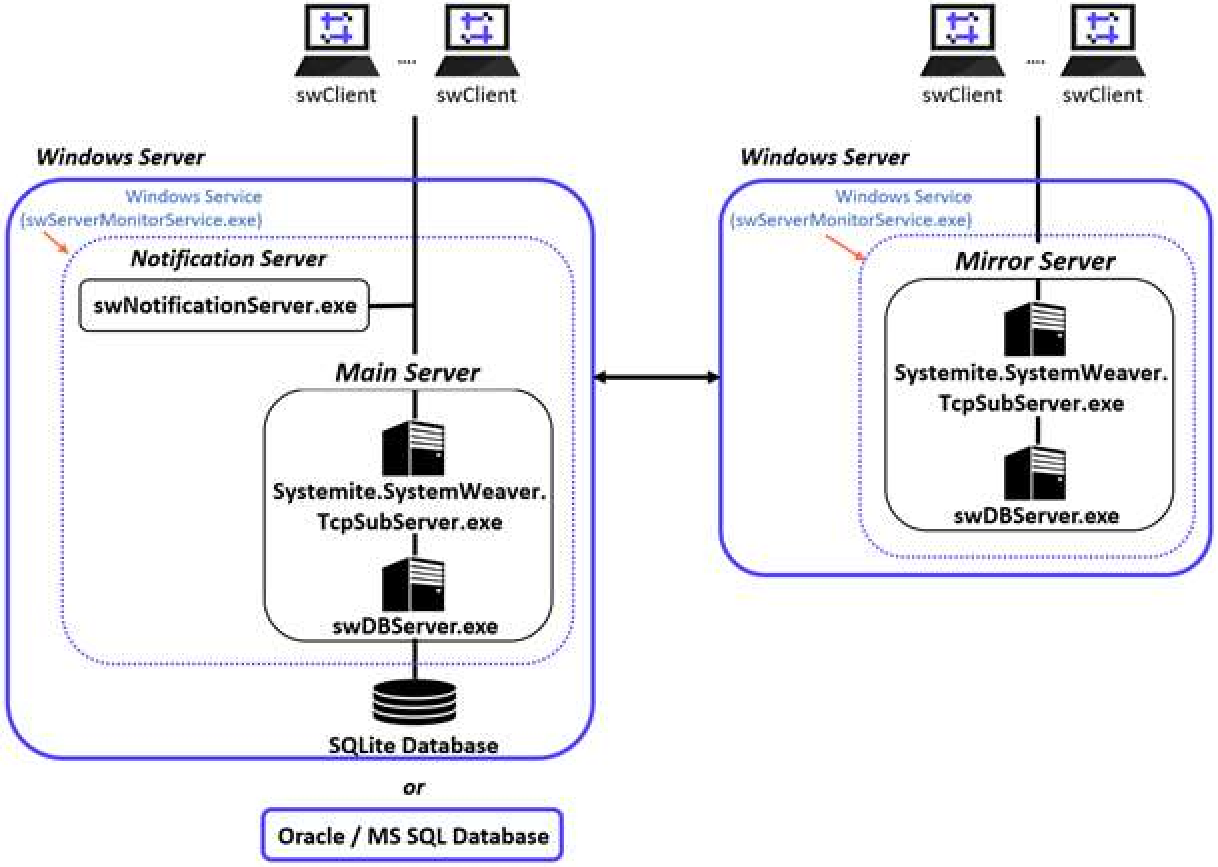
Including a swNotificationServer in the configuration means that notification emails can be distributed automatically whenever:

- The status or assignment of an *issue* is changed.
- A note is added to an issue.
- An assignee is added to a *Request for status change*.

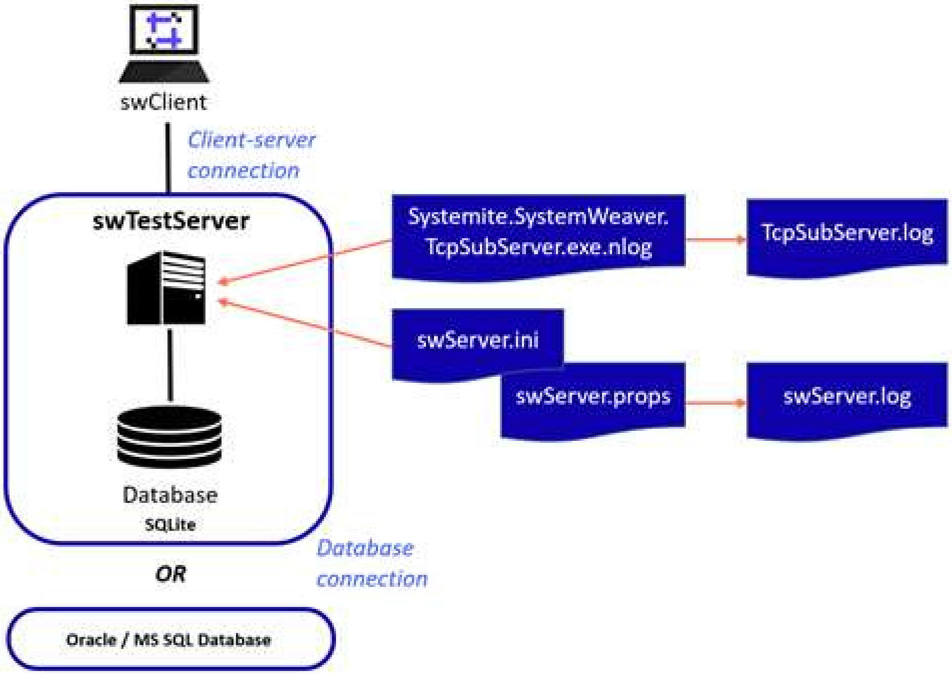


2.3.3 Full Configuration

The figure below shows a sample SystemWeaver configuration including a main server, one mirror server, and a notification Server.



2.4 SystemWeaver Server



All SystemWeaver servers, like the swDBServer or the swTestServer server applications use the 'swServer.ini' configuration file, see Format of swServer.ini file. This file name must normally be used, except for the following cases:

- When the server is started by a swServerMonitorService, when the file name is specified in the 'swServerMonitorService.settings' file. See Format of swServerMonitorService.settings file,
- When the swDBServer is started on a Windows command line, where the parameter 'ini' can be used for specifying a different file name and path.

2.4.1 swTestServer

The swTestServer offers the same functions as an swDBServer, but with a Windows GUI. The GUI makes it easier to test and observe an installation, although the GUI excludes the execution as a Service.

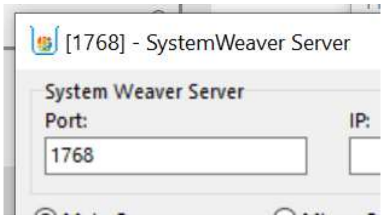
The GUI of the swTestServer is illustrated in the figures below.

The application icon indicates if it is started or not.

Indicates the test server is stopped

Indicates the test server is running

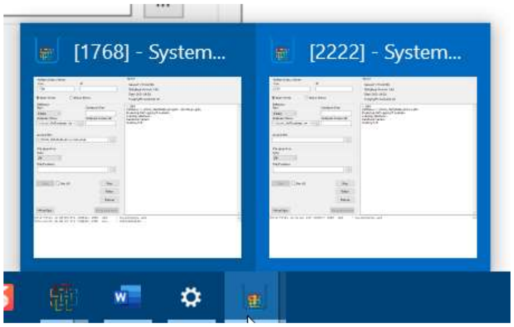
The server port number is displayed in the swTestServer GUI caption, which also displays when hovering over the executable on the Windows taskbar.



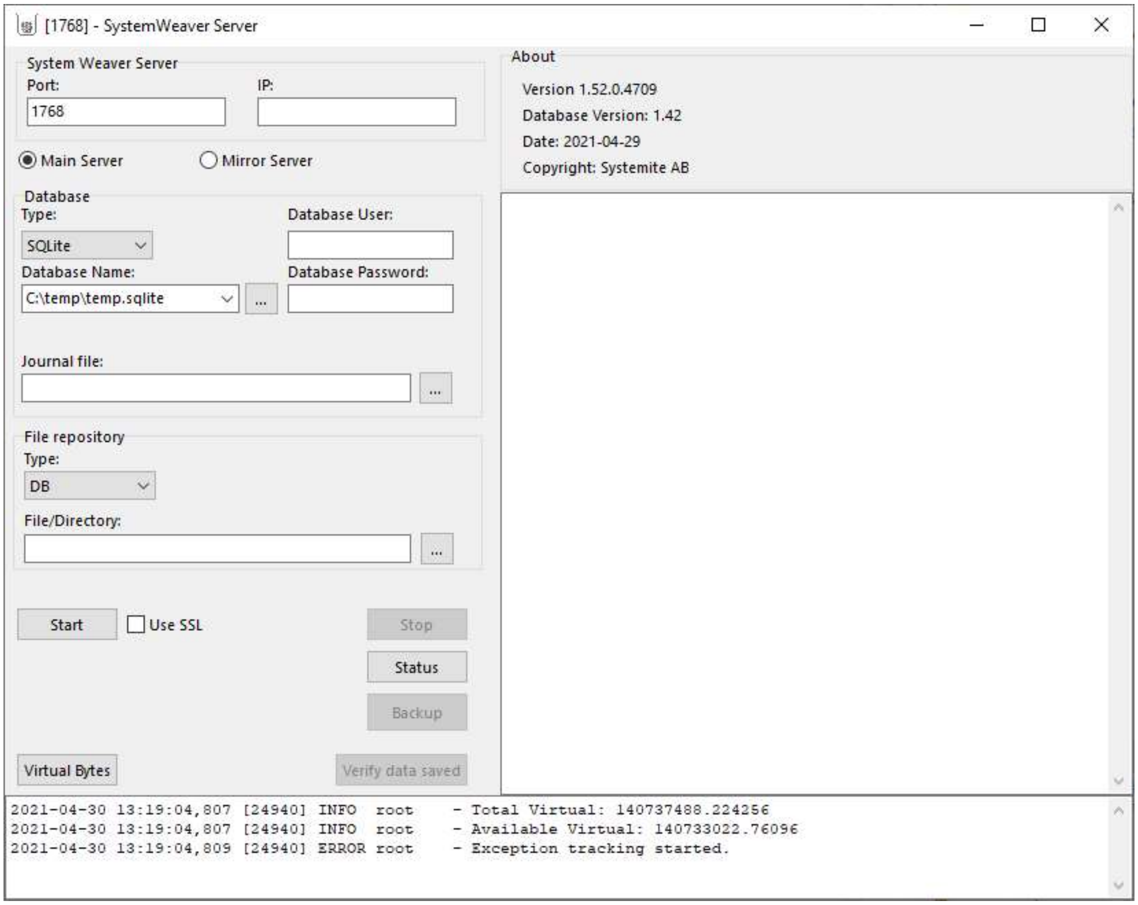
[1768] - SystemWeaver Server

System Weaver Server

Port: 1768



GUI of Main Server in swTestServer



[1768] - SystemWeaver Server

System Weaver Server

Port: 1768

IP:

● Main Server ○ Mirror Server

Database

Type: SQLite

Database User:

Database Name: C:\temp\temp.sqlite

Database Password:

Journal file:

File repository

Type: DB

File/Directory:

☐ Use SSL

About

Version 1.52.0.4709

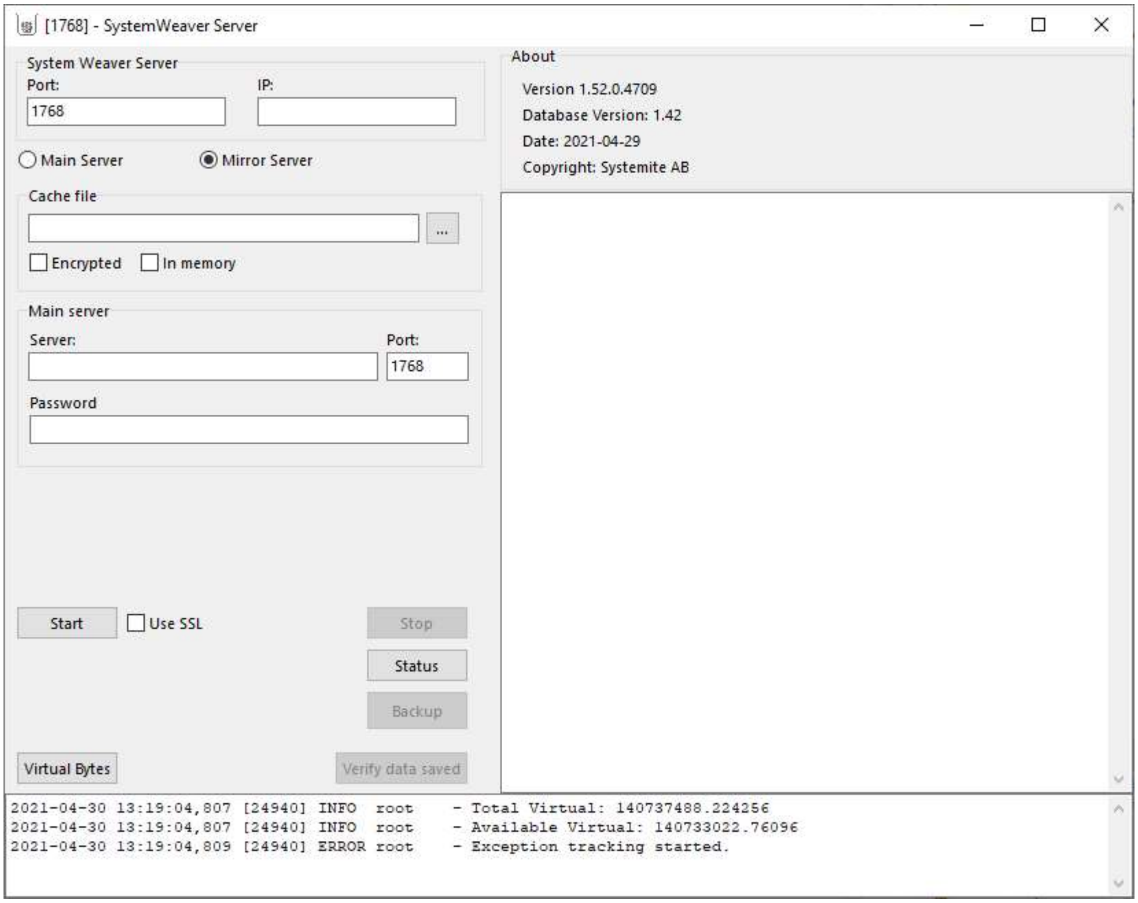
Database Version: 1.42

Date: 2021-04-29

Copyright: Systemite AB

```
2021-04-30 13:19:04,807 [24940] INFO  root    - Total Virtual: 140737488.224256
2021-04-30 13:19:04,807 [24940] INFO  root    - Available Virtual: 140733022.76096
2021-04-30 13:19:04,809 [24940] ERROR root    - Exception tracking started.
```

GUI of Mirror Server in swTestServer



Port: 1768

IP:

○ Main Server ● Mirror Server

□ Encrypted □ In memory

Server:

Port: 1768

Password

□ Use SSL

Version 1.52.0.4709
Database Version: 1.42
Date: 2021-04-29
Copyright: Systemite AB

```
2021-04-30 13:19:04,807 [24940] INFO  root    - Total Virtual: 140737488.224256
2021-04-30 13:19:04,807 [24940] INFO  root    - Available Virtual: 140733022.76096
2021-04-30 13:19:04,809 [24940] ERROR root    - Exception tracking started.
```

The entries that are possible to modify using the GUI are described in Format of swServer.ini file.

The other operations available in the GUI are described below:

Start Starts the server. The server first performs initialization and checks the integrity of the database. After the initialization, the server is ready to accept connections from SystemWeaver clients.

Stop Stops the server. The server first safely disconnects all connected clients.

Status Prints the current status details for the server.

Backup Makes a back-up copy of SQLite databases.

Virtual Bytes Prints the current virtual memory used by the server.

Verify data saved Checks that the content of the main memory database corresponds to the SQL database. The operation is intended for testing purposes, as the main memory database should always be in sync with the SQL database.

Note that the swTestServer uses the .ini file for initial settings, if there are any.

2.4.2 swDBServer

The swServer can either be executed in the foreground by double-clicking the application icon, or from a Windows command line:

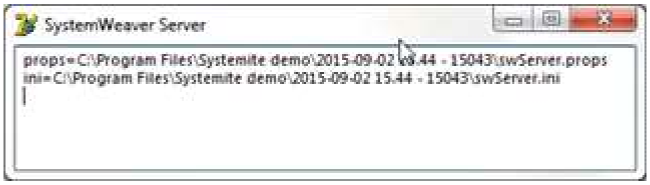
Example:

```
swDBServer.exe ini="C:\SystemWeaver\Server\swServer.ini" props="C:
\SystemWeaver\Server\swServer.props"
```

The recommended way is in the background as a Windows service (see section SystemWeaver as a Service for details).

If the swServer is started in the foreground, it displays a simple status window:

GUI of swServer



```
props=C:\Program Files\Systemite demo\2015-09-02 15.44 - 15043\swServer.props
ini=C:\Program Files\Systemite demo\2015-09-02 15.44 - 15043\swServer.ini
```

The swServer is stopped gracefully by clicking the exit button ☒.

2.4.3 Command line options

The following command line options are available for the swServer:

`Ssl` [true/false]Same function as the *UseSSL* entry of the .ini file, refer to Format of swServer.ini file.*

* The availability of the encryption option is restricted due to EU regulations. Please contact us for more information.

2.4.4 Run-time methods

The swServer supports the following run time methods:

TBD (backup)

2.5 SystemWeaver as a Service

SystemWeaver Servers (SystemWeaver Server) can be executed as Windows Services using the *swServerMonitorService* utility application. This is recommended since it offers greater flexibility and additional automatic restart functionality of the managed processes.

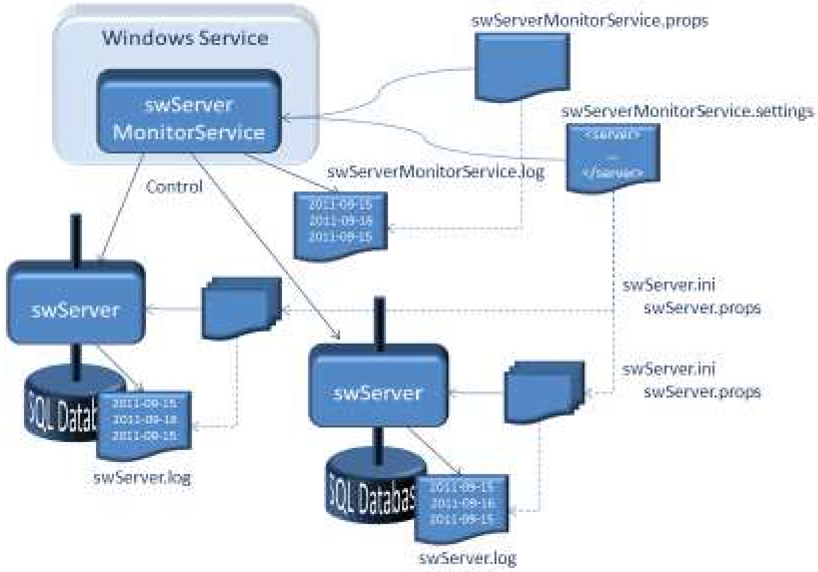


Figure - swServerMonitorService

swServerMonitorService

Running the SystemWeaver server application as a Windows Service offers the following advantages:

- The Service and server application can be started automatically when the server computer is started.
- The server application can optionally be restarted automatically (see Format of swServerMonitorService.settings file) in the case it has been terminated, e.g., due to a manual termination or server computer failure.
- The server does not depend on a login session, i.e., no user needs to be logged in to the server computer.
- The service offers slightly higher performance since no on-screen logging is performed.

Before using a service installation, you should make sure that the involved server applications can be executed correctly as normal applications, using the same versions and setting files. You can either test with the swTestServer, or the swServer (this may be called swDBServer depending on the version) executed in the foreground. If there are any problems with the configurations, these will be more difficult to analyze when the servers are run as services.
A swServerMonitorService installation involves the following resources in addition to the SystemWeaver database(s):

swServerMonitorService_install.bat The batch file that installs the SystemWeaver Monitor Service as a service on the server computer.

swServerMonitorService_uninstall.bat The batch file that uninstalls the SystemWeaver Monitor Service as a service on the server computer.

swServerMonitorService.exe The service application that manages all SystemWeaver server applications on a server computer. It starts all the server applications automatically and can restart them when necessary.

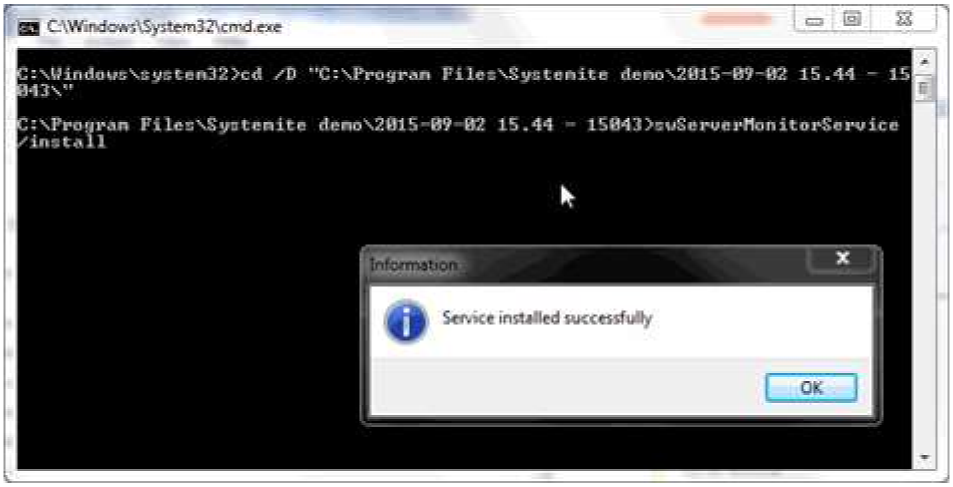
NLog.config The file used for configuration of the logging function. A sample is normally supplied in the SystemWeaver installation package. The file format is described in Format of swServerMonitorService.exe.nlog file.

swServerMonitorService.settings The Settings file defines all the server applications and their input files that are run under the service. A sample .settings file is normally supplied in the

	SystemWeaver installation package. The file format is described in Format of swServerMonitorService.settings file.
swServerMonitorService_debug.bat	The batch file installs the SystemWeaver Service Monitor in debug mode. This should only be used for troubleshooting as the logging will decrease performance to some extent.

Follow the steps below in order to set up a swServerMonitorService installation:

1. Select an installation folder to store the files of the SystemWeaver service package (the files listed above). The folder should preferably be on a local drive to minimize the dependency on other servers.
2. Edit the .props and .settings files according to the needs of the installation (refer to the paragraphs Format of .props file and Format of swServerMonitorService.settings file for a description of the files), especially all file paths, which could, e.g., be set to the installation folder to minimize the risk for mixing up files in case multiple server applications are executed on the same computer.
3. Install a SystemWeaver server monitor by running the swServerMonitorService_install.bat (in a Windows Explorer window). This will open a command prompt window that should be closed when this is prompted for by the window:



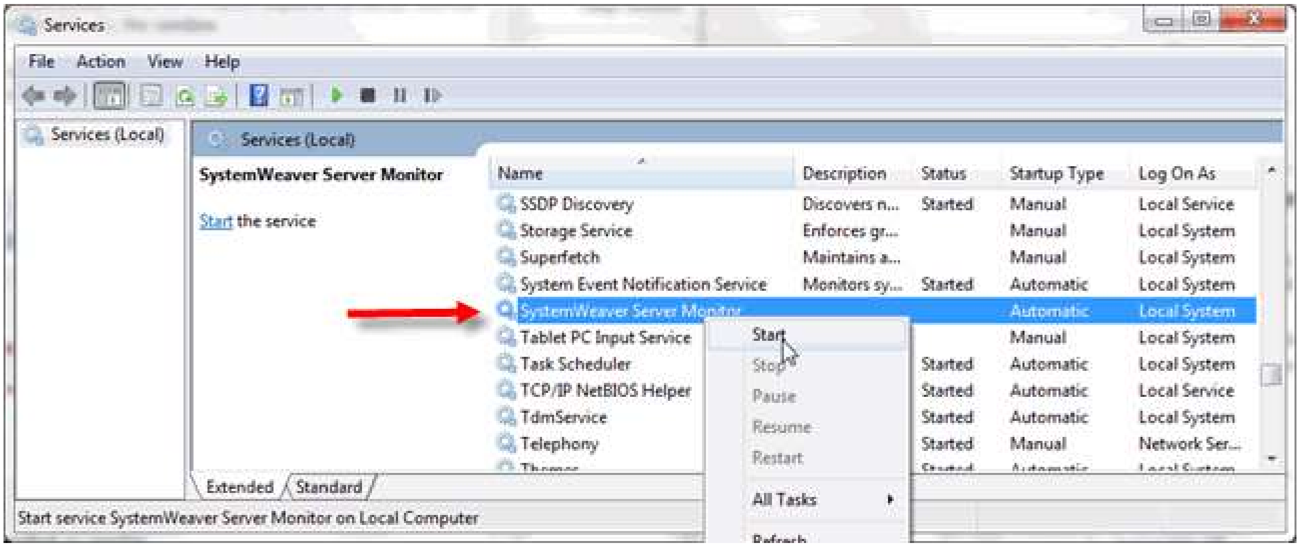
```
C:\Windows\system32>cd /D "C:\Program Files\Systemite demo\2015-09-02 15.44 - 15
043\"

C:\Program Files\Systemite demo\2015-09-02 15.44 - 15043>swServerMonitorService
/install
```

Information

Service installed successfully

4. Open the Services tool in the Windows Control Panel.
   (The Services tool can normally be found under the Control Panel "Administrative Tools" subfolder.)
   As can be seen below, the Startup Type is Automatic, but the Status is still not Started, since a restart of the host computer is required, or the use of the Start command:



SystemWeaver Server Monitor

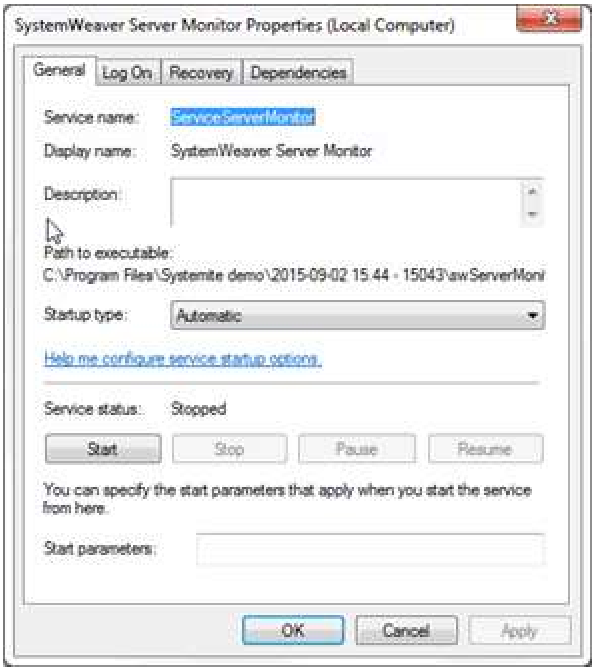
Start the service

Name	Description	Status	Startup Type	Log On As
SSDP Discovery	Discovers n...	Started	Manual	Local Service
Storage Service	Enforces gr...		Manual	Local System
Superfetch	Maintains a...		Manual	Local System
System Event Notification Service	Monitors sy...	Started	Automatic	Local System
SystemWeaver Server Monitor			Automatic	Local System
Tablet PC Input Service			Manual	Local System
Task Scheduler		Started	Automatic	Local System
TCP/IP NetBIOS Helper		Started	Automatic	Local Service
TdmService		Started	Automatic	Local System
Telephony		Started	Manual	Network Ser...

Start service SystemWeaver Server Monitor on Local Computer

Figure - SystemWeaver Server Monitor installed as service

5. Review the properties of the SystemWeaver Server Monitor, see the figure below.


SystemWeaver Server Monitor Properties (Local Computer)
General
Log On
Recovery
Dependencies
Help me configure service startup options.
Start
Stop
Pause
Resume
OK
Cancel
Apply

Service name: ServiceServerMonitor

Display name: SystemWeaver Server Monitor

Description:

Path to executable:
C:\Program Files\Systemite demo\2015-09-02 15.44 - 15043\swServerMoni

Startup type: Automatic

Service status: Stopped

You can specify the start parameters that apply when you start the service from here.

Start parameters:

Figure - SystemWeaver Server Monitor service properties

- The PATH TO EXECUTABLE setting is made automatically.
- The STARTUP TYPE should normally be set to "Automatic".
- No START PARAMETERS are needed, as the server gets all its settings from the .props and .settings files.
- The log on settings under the LOG ON tab should remain to log on using the LOCAL SYSTEM account.
- The settings under the RECOVERY tab can also remain unchanged, i.e., no automatic recovery of the Server Monitor is normally needed.
- The Server Monitor has no dependencies to other services, so the DEPENDENCIES tab should also remain

unchanged.

The service, when started, can be observed in the Windows Task Manager under the "SYSTEM" User Name:

Note that you may need to tick the Show processes from all users option, since the process is run under the SYSTEM user.

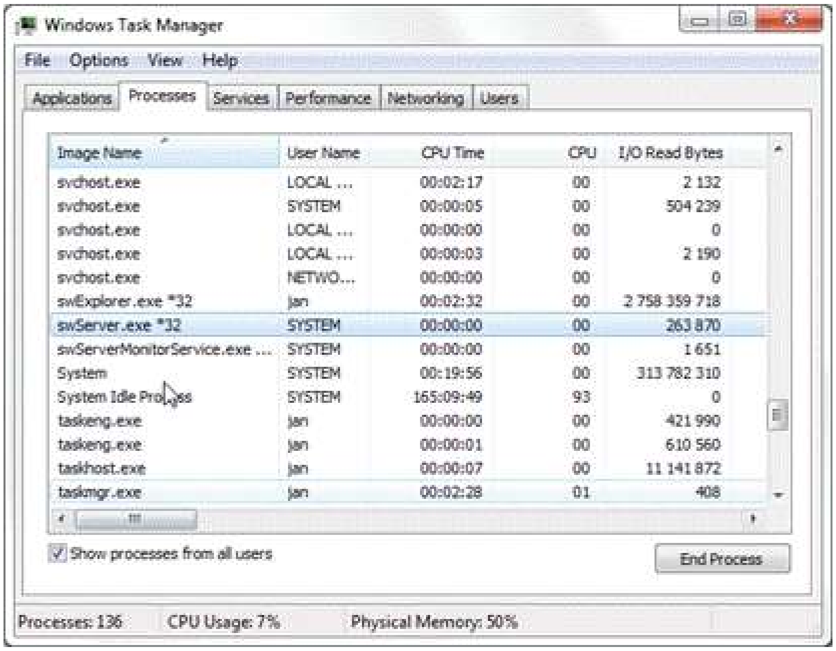


Image Name	User Name	CPU Time	CPU	I/O Read Bytes
svchost.exe	LOCAL ...	00:02:17	00	2 132
svchost.exe	SYSTEM	00:00:05	00	504 239
svchost.exe	LOCAL ...	00:00:00	00	0
svchost.exe	LOCAL ...	00:00:03	00	2 190
svchost.exe	NETWO...	00:00:00	00	0
swExplorer.exe *32	jan	00:02:32	00	2 758 359 718
swServer.exe *32	SYSTEM	00:00:00	00	263 870
swServerMonitorService.exe ...	SYSTEM	00:00:00	00	1 651
System	SYSTEM	00:19:56	00	313 782 310
System Idle Process	SYSTEM	165:09:49	93	0
taskeng.exe	jan	00:00:00	00	421 990
taskeng.exe	jan	00:00:01	00	610 560
taskhost.exe	jan	00:00:07	00	11 141 872
taskmgr.exe	jan	00:02:28	01	408

☑ Show processes from all users

Processes: 136 CPU Usage: 7% Physical Memory: 50%

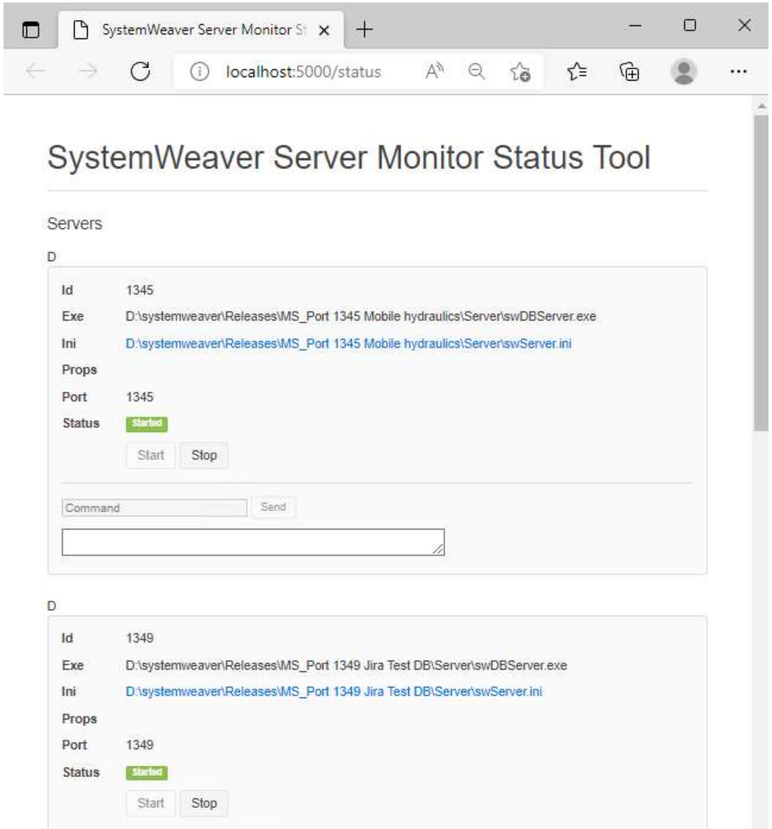
Figure - SystemWeaver Server Monitor service properties

See Server Monitor Service Status Tool to configure an optional web app tool that can be used to stop/start individual server application(s) run by the service.

2.5.1 Server Monitor Service Status Tool

The *SystemWeaver Server Monitor Service Status Tool* is a web application that can be used to stop and start any server application that is running as part of the *SystemWeaver Monitor Service*. It is optional, and only beneficial to use if there is a need in your operations and maintenance routines to stop one or more, and not all, server applications that are running in the service.

The tool can be run directly as an executable in a Windows session on an as-needed basis, or as a server application in the service. It is only accessible from the server machine where the *SystemWeaver Monitor Service* is running.



SystemWeaver Server Monitor Status Tool

Servers

D

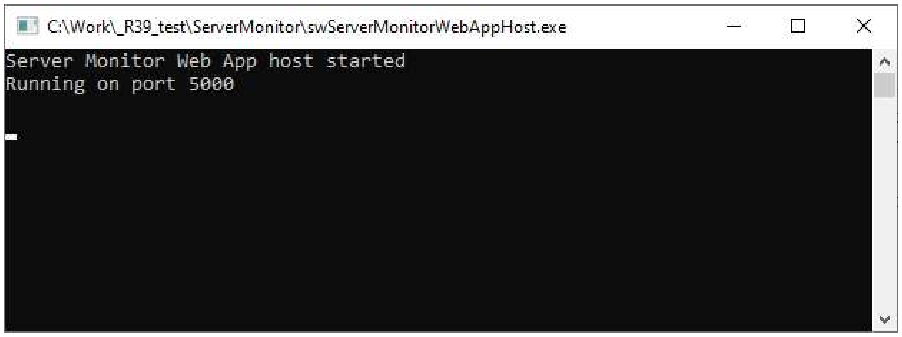
Id	1345
Exe	D:\systemweaver\Releases\MS_Port 1345 Mobile hydraulics\Server\swDBServer.exe
Ini	D:\systemweaver\Releases\MS_Port 1345 Mobile hydraulics\Server\swServer.ini
Props	
Port	1345
Status	Started

D

Id	1349
Exe	D:\systemweaver\Releases\MS_Port 1349 Jira Test DB\Server\swDBServer.exe
Ini	D:\systemweaver\Releases\MS_Port 1349 Jira Test DB\Server\swServer.ini
Props	
Port	1349
Status	Started

The configuration of the Status Tool is defined in the swServerMonitorWebAppHost.exe.config file, which is described in Format of swServerMonitorWebAppHost.exe.config.

When the service is running and the Status Tool is configured, you can run swServerMonitorWebAppHost.exe directly by either double-clicking on it or right-clicking and using **Run as administrator**.



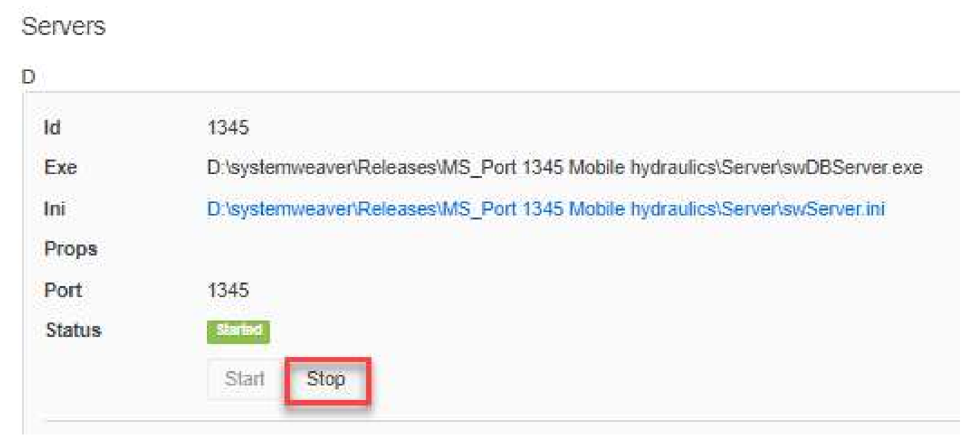
C:\Work_R39_test\ServerMonitor\swServerMonitorWebAppHost.exe

```
Server Monitor Web App host started
Running on port 5000
```

Alternatively, you can run the Status Tool as a Server in the service.

Once it is running, you can view the status of each server application in the service by opening a browser on the machine where the service is running and navigating to http://localhost:5000/status (i.e., the port here should match the one in the *key*="WebHostPort" in swServerMonitorWebAppHost.exe.config). "5000" is only an example.

To stop a specific server in the service, find the entry in the tool and click **Stop**.



Servers

D

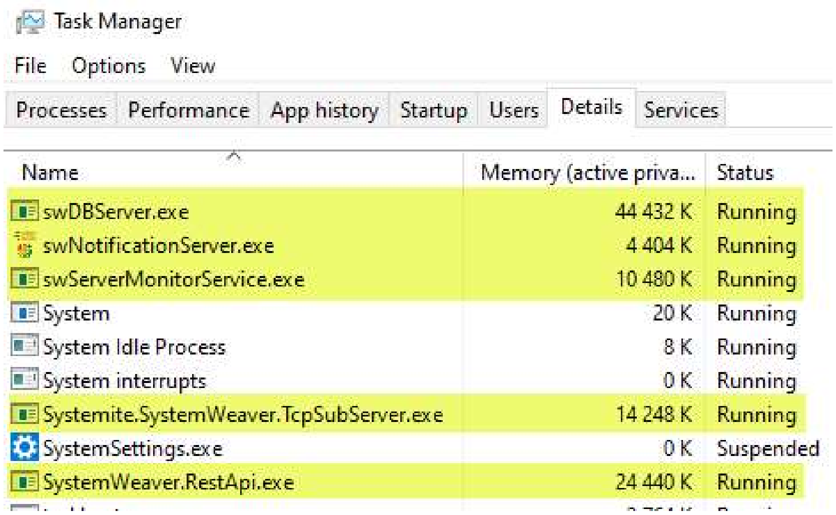
Id	1345
Exe	D:\systemweaver\Releases\MS_Port 1345 Mobile hydraulics\Server\swDBServer.exe
Ini	D:\systemweaver\Releases\MS_Port 1345 Mobile hydraulics\Server\swServer.ini
Props	
Port	1345
Status	Started

When you are ready to restart the server, click **Start** in the tool.

Note that if the database is very large, it may take some time to complete the stop and start processes. You can monitor the progress via the Task Manager.

- **Stop:** It is completely stopped once the server application is no longer listed as a running process.
- **Start:** It has completed the startup process once the Memory usage has stopped growing.

Example:



Name	Memory (active priva...	Status
swDBServer.exe	44 432 K	Running
swNotificationServer.exe	4 404 K	Running
swServerMonitorService.exe	10 480 K	Running
System	20 K	Running
System Idle Process	8 K	Running
System interrupts	0 K	Running
Systemite.SystemWeaver.TcpSubServer.exe	14 248 K	Running
SystemSettings.exe	0 K	Suspended
SystemWeaver.RestApi.exe	24 440 K	Running

Note: If the Status tool is left open and inactive, you will need to refresh (F5) the page to use it.

2.6 SystemWeaver Notification Server

A SystemWeaver Notification Server is used for sending notification emails to SystemWeaver users (and others) at certain events, like when an *issue* has been assigned to a user. ([1] SystemWeaver Explorer Help Manual)

The notification server listens for events from the SystemWeaver server, picks out those that should result in a notification email, reads additional information required for the notification message from the server, and sends an email message to an SMTP server.

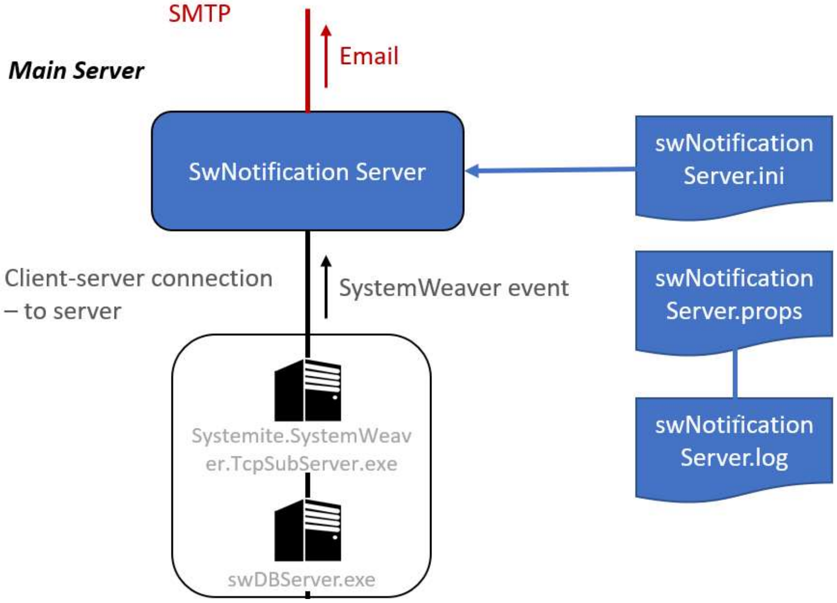

SMTP
Email
Main Server
SwNotification Server
swNotification Server.ini
Client-server connection – to server
SystemWeaver event
swNotification Server.props
swNotification Server.log
Systemite.SystemWeaver.TcpSubServer.exe
swDBServer.exe

Figure - swNotificationServer

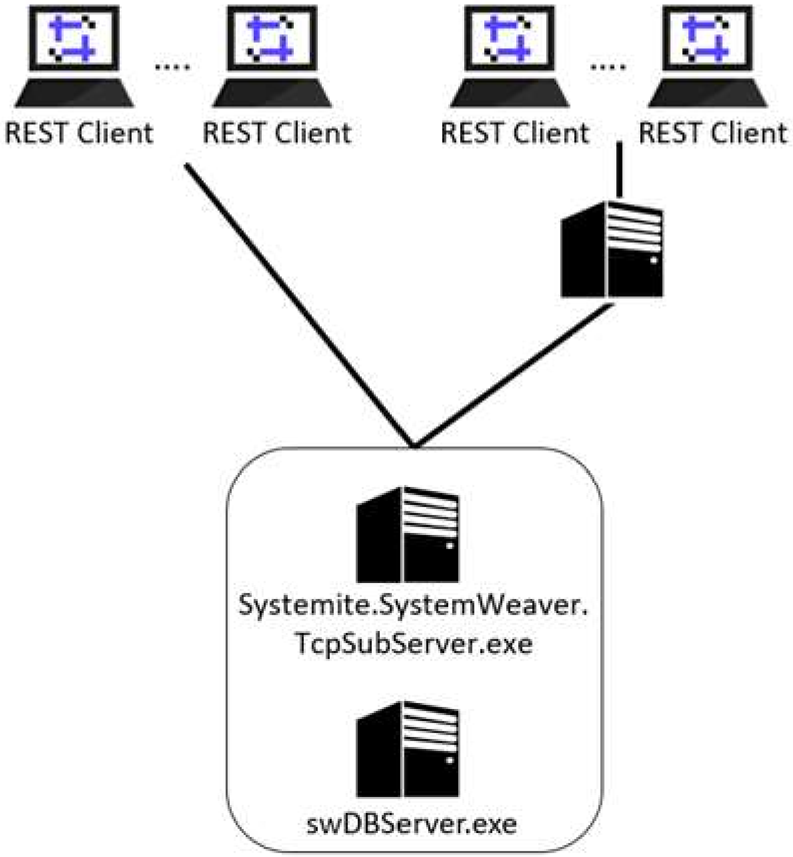
The configuration of the notification server is defined in the swNotificationServer.ini file, which is described in Format of swNotificationServer.ini file.

When setting up the configuration it can be practical to test the configuration parameters, using the swTestNotificationServer utility.

2.7 SystemWeaver REST API Server

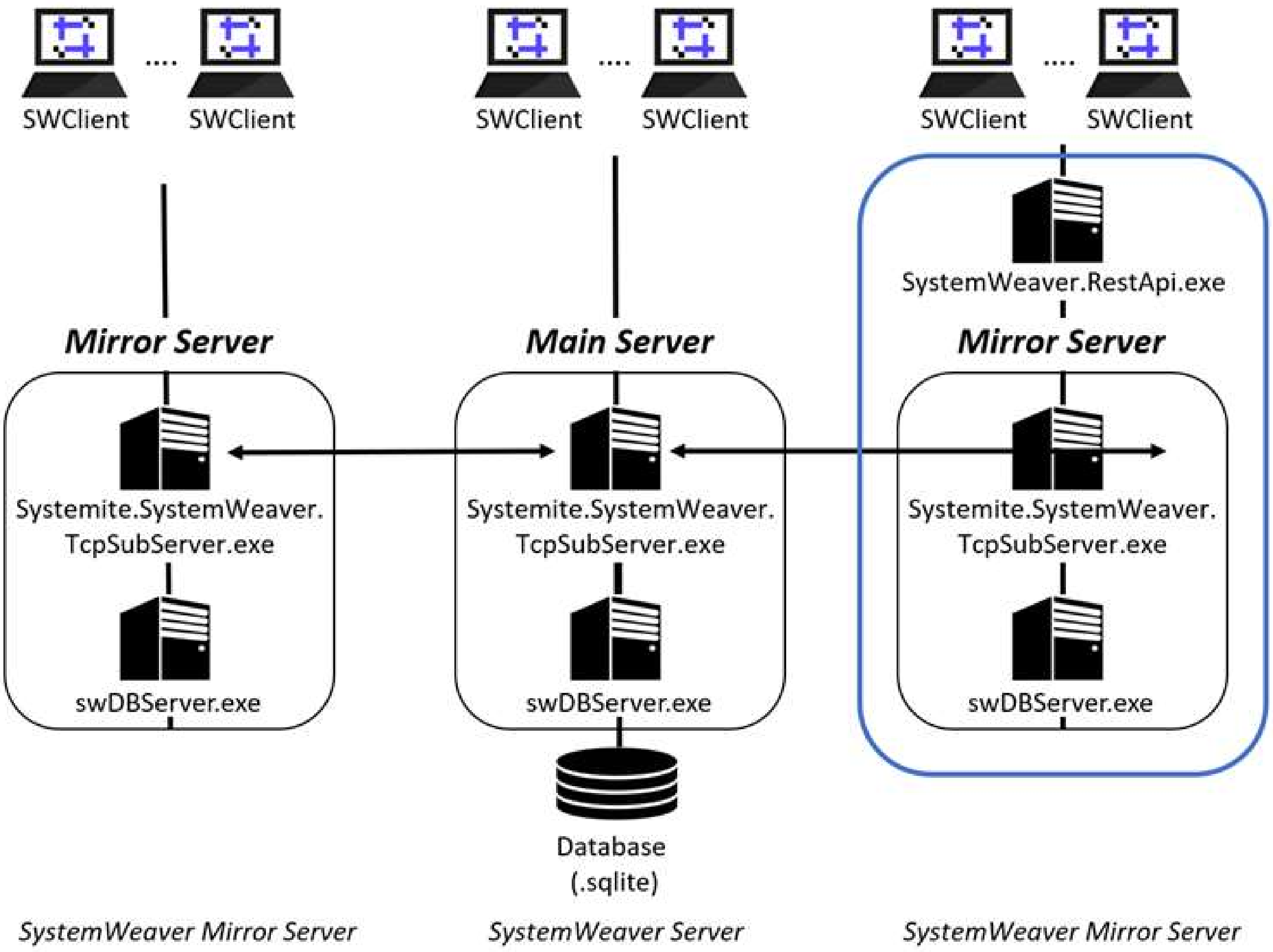
The SystemWeaver REST API Server allows users to connect to a SystemWeaver server or mirror server through an HTTP interface, enabling better support for integration with tools and applications on non-Windows platforms.

It is installed as an independent executable, connecting to the server Systemite.SystemWeaver.TcpSubServer.exe in a similar way as a standard client, as shown in the picture below. The SystemWeaver REST API Server is not required to be installed on the same machine as the SystemWeaver server or mirror server that it is connected to, but for performance reasons it is highly recommended, given that the amount of available RAM is sufficient.



*SystemWeaver Server/
SystemWeaver Mirror Server*

It is possible to allow both standard SystemWeaver Clients and REST Clients to connect to the same SystemWeaver server or mirror server. However, for a production environment it is recommended to have one or more mirror servers dedicated for REST API communication, as shown below. This will minimize performance impact on SystemWeaver Client users. Apart from this, the SystemWeaver REST API server follows the same scaling principles as a mirror server.



For information on how to install, see Installing the SystemWeaver REST API Server.

2.8 Configuration files

2.8.1 General conventions

Location of files

Unless otherwise stated, configuration files are assumed to be located in the same folder as the .exe file that is being configured.

Specifying values

For the file types described below, there is no need to enclose values in quotation marks. Each value must however be on a single line.

Example:

```
MailServerLoginName=SystemWeaver Notification Service
```

Note on security:

Some of the file types contain passwords. This means that only the Administrator should have 'Read' rights to these files where they are stored in the file system. There is also an option in SystemWeaver to store the password in encrypted format for the cases when the password is sent unencrypted on the network. All communication between SystemWeaver applications is based on SSL, which means that passwords are always encrypted on the network. However, for the communication between a SystemWeaver server and a database server the password may be sent unencrypted in the database communication protocol. In this case it is preferred to use encrypted passwords. The SystemWeaver utility tool 'EncryptString.exe' can be used for this purpose.

2.8.2 Specifying .ini files

The .ini file should by default have the file name listed below:

Application	**Default ini File name**
swDBServer (for main and mirror servers)	swServer.ini (with format according to Format of swServer.ini file)
swNotificationServer	swNotificationServer.ini (with format according to Format of swNotificationServer.ini file)

For the swDBServer, the .ini file could optionally be specified by the command line parameter 'ini' where the file name and path must be complete:

```
swServer ini="C:\myinifile.ini"
```

When a swDBServer is run by a swServerMonitorService, the ini file is specified by the <ini> entry in the .settings file, as defined in section Format of swServerMonitorService.settings file

2.8.3 Specifying .props files

Purpose

The .props file should by default have the file name listed below: (For the format of the .props file refer to Format of .props file)

Application	**Default props File name**
swDBServer	swServer.props
swExplorer*	swExplorer.props

For the swDBServer, the .props file could optionally be specified by the command line parameter 'props' where the file name and path must be complete:

```
swServer props="C:\mypropsfile.props"
```

When a swDBServer is run by a swServerMonitorService, the props file is specified by the <props> entry in the .settings file, as defined in section Format of swServerMonitorService.settings file

* Note that normally no logging is needed for the swExplorer, since any error messages become visible to the user, so this file can

safely be omitted from a configuration.

2.8.4 Format of swServer.ini file

The swServer.ini file has the following content: (values to the right of the equal sign are typed in gray and are examples only to illustrate the used format)

```
[MainServer]
dbType=SQLite
DSN=C:\temp\temp.sqlite
;JournalFile=C:\temp\temp.swjournal
MinJournalEntries=
SyncJournalEntries=
FileRepository=C:\FileRepos
FileRepositoryType=Files
DatabaseUser=
DatabasePassword=
DatabasePasswordEncrypted=
CallLogging=on/detailed
LogFileDirectory=
AllowLocalCache=
[Tcp]
Port=1768
IP=
CertPassword=aaaa
AutoStart=False
DLQ=False
DLQFileDirectory=
DLS=False
[MirrorServer]
;MainServer=localhost
;MainServerPort=1768
;MainServerPassword=password
;MainServerPasswordEncrypted=
;Cache=D:\temp.cache
;CacheEncrypted=
;CacheInMemory=
```

The entries of the file are described below.

The **MainServer** section contains the settings for the main server.
The **Tcp** section contains the settings for communication between the main server and the client applications.
The **MirrorServer** section contains the settings for the *mirror server*, if applicable (see Mirroring).

The **dbType** entry defines what type of database is being used. The following options are supported:

SQLite - SQLite databases
MS SQL Server - SQL Server databases
Oracle - Oracle databases

The **DSN** entry defines the file path to the database in case of file based database. (SQLite). Note: The file path should not be written with quotation marks or apostrophes, even for paths including blank (space) characters.
For Oracle, the DNS should be the TNS alias of the Oracle server. Refer to Oracle documentation for details on TNS.

The **JournalFile** entry is used to enable *journaling* (refer to Mirroring for information). It should define the file path to and file name of the SystemWeaver .swjournal file. Journaling has to be enabled on the main server when using mirroring so that the mirror server(s) can fetch updates since the last runtime.

The **MinJournalEntries** entry defines the minimum number of entries stored in the .swjournal file. The SystemWeaver Server will remove old entries according to this target limit. If left unspecified, the .swjournal file will keep growing with each update of the database.

SyncJournalEntries allows the main server to run transactions found in the journal file that are not found in the database upon start-up. During normal operation on a local disk, there is no risk of this scenario occurring. However, this option does provide an automated way to handle an unforeseen synchronization problem. We do not recommend the use of SyncJournalEntries by default, since it can potentially hide network/disk problems that you otherwise would want to be aware of.

The **FileRepository** entry defines the file path to the file repository. If the FileRepositoryType is 'Files' the path should identify the folder under which the files will be stored. If the type is 'DB' the path defines the SQLite database file. (Currently SQLite is the only supported database file repository)

The **FileRepositoryType** defines the type of file repository. Possible values are 'DB' or 'Files'.

DatabaseUser is the user name for logging in to the database, if required. This entry can be left empty in case no login is required.

DatabasePassword is the password used for login to the database, if required. This entry can be left empty in case no login is required.
Note that this password is not a SystemWeaver user password.

DatabasePasswordEncrypted is an encrypted password used for login to the database, if required. An example is when the SystemWeaver Server is located on one physical server and the database server on another*, and security policies prohibit that unencrypted passwords is sent on the network between the severs. This entry may be left empty in case no login is required. The SystemWeaver tool EncryptString.exe may be used for generating an encrypted password for this purpose.

CallLogging is an entry that turns on extended performance logging in the server. When logging is activated a new log file is generated each time the server is started, with automatically defined file names. Alternatives are 'on' for standard logging, 'detailed' for extended logging, and none (or any invalid entry) for no logging at all.

LogFileDirectory defines the file path to the CallLogging log file. If left blank, the file will generate in the same directory as the server executables.

AllowLocalCache defines whether or not connecting mirror servers can use a local cache file or must use in-memory cache. Supported values are True or False. True is the system default. If AllowLocalCache is set to False, mirror servers connecting to the main server can only run using in-memory cache (CacheInMemory=True in mirror server swServer.ini).

The **Port** entry defines the TCP/IP port number to be used by the server. Refer to the section about IP Address and Ports for details.

IP is the IP address or domain name to be used by the SystemWeaver server. This entry should only be used for the case that the server uses multiple IP addresses or domain names.

The **CertPassword** is used for authentication purposes in the SSL (Secure Socket Layer), so that the SystemWeaver clients can detect the legitimacy of the server. However for intranet use, authentication is normally not required and SystemWeaver is therefore distributed with dummy certificates, with the dummy password "aaaa". If there is a need for specific certificates, the certificate files in the "cert" folder can be replaced by certificates generated by OpenSSL (www.openssl.org).

The **AutoStart** option with the possible values 'True' and 'False' enables the swTestServer to start and connect to the database immediately after the server application is started. The value 'True' should normally be used only after the server configuration has been verified to be fully functional. The swDBServer.exe requires 'True' since this is the only way to start the server.

UseSSL* selects whether SSL should be used or not. Alternatives are 'True' for SSL and 'False' for not using SSL. Within a closed network 'False' could be used to improve robustness for network disturbances, while 'true' should be used for communications over Internet.

DLQ defines whether a .dlq file should be generated (True) or not (False) when a split server architecture is used. The default setting is False, but it is recommended that it be set to True, at least initially, since its content allows for proper analysis should there be issues with server performance. IT administrators should be aware that a .dlq file can take some space depending on the size and activity level in the server (e.g., 10GB for very large and active systems).

DLQFileDirectory defines the file path to the .dlq file. If left blank, the file will generate in the same location as the server executable. Note that the system account running the server must have write permissions to the directory being written to and the directory must exist. Otherwise, the server will not start.

DLS used for high-frequency realtime data logging for analysis purposes. The default setting is no logging (False).

MainServer defines the domain address of the SystemWeaver main server (swServer.exe) that the mirror server should connect to. The address may be specified as either a fixed numerical address (123.456.789.012) or a domain name (EV198).

MainServerPort defines the port number of the SystemWeaver main server that the mirror server should connect to. It should match the **Port** entry in the **Tcp** section above. Refer to the section about IP Address and Ports for details.

MainServerPassword defines the SystemWeaver password used to run the main server. It should be set to the password of the "system" user account.

MainServerPasswordEncrypted is an alternative to the *MainServerPassword* entry, where the password is entered in encrypted form. The SystemWeaver utility tool EncryptString.exe may be used for generating an encrypted password for this purpose.

Cache defines the file path to the local .cache file used by the mirror server.

CacheEncrypted* is an option to encrypt the cache file. The default setting is no encryption (False).

CacheInMemory is an option to run a mirror server without having a local cache file. Only suitable for databases smaller than 1 GB, or in a QA environment where you have less requirements on the server uptime. We always recommend running the mirror with a local cache file.

* Installing a SystemWeaver server and the database server on separate computers may affect access performance, depending on the network performance between these computers. The availability of the encryption option is restricted due to EU regulations. Please contact us for more information.

2.8.5 Format of swNotificationServer.ini file

The swNotificationServer.ini file has the following content:

```
SystemWeaverServerLocation=
SystemWeaverServerPort=
SystemWeaverLoginName=
SystemWeaverPassword=
MailServerName=
MailServerPort=25
MailServerLoginName=
MailServerLoginPassword=
MailServerFromName=
MailServerFromAddress=
MailServerReplyToName=
MailServerReplyToAddress=
UseTLS=[None|TLS_1|TLS_1_2]
ConfigMessage=
```

The different entries are described below.

SystemWeaverServerLocation is the IP-number or domain name of the computer where the SystemWeaver server is running. Note that when working with a local server installed on the same PC as the swNotificationServer, then the logical IP domain name "localhost" may be used in the swExplorer in order to specify the local server.

SystemWeaverServerPort is the port number that the SystemWeaver server is using (see Format of swServer.ini file).

SystemWeaverLoginName is the user name that the swNotificationServer uses for logging in to the SystemWeaver server. Note: in order to guarantee that the notification server will get access and be able to manage any project of the server, the special user account 'system' should be used. Refer to [2] SystemWeaver Access Rights and Administration , for details on predefined user accounts like 'system'.

SystemWeaverPassword is the password of the SystemWeaverLoginName account.

MailServerName selects the SMTP server to be used for sending the emails. Example: "mailhost.example.com".

MailServerPort is the port number used for the connection to the SMTP server. 587 is the usual port when authentication is used, and 25 when no authentication is used. See also the note on the *UseTLS* entry.

MailServerLoginName defines the account name to be used for the mail server connection, if authentication is used (see the *UseTLS* entry).
If no authentication is used, the entry can be left empty.
The MailServerLoginName, or an email address associated with the account, will appear in the "From" field of the sent notification emails, unless the *MailServerFromAddress* entry is used. (Many email servers (POP3/IMAP) will use the *MailServerLoginName* even if the *MailServerFromAddress* is specified.)
Example: "NotificationService@example.com".
Note that the *MailServerLoginName* must include a valid domain name, like @example.com.

Technically, this entry will be included in the *From:* field of the notification email.

MailServerLoginPassword defines the password corresponding to the *MailServerLoginName* account.
If no authentication is used, the entry can be left empty.

MailServerFromName defines the 'display name' of the sender, presented as the sender's name, in the receiver's email client.
Note that, for authenticated email transmissions, there is a chance that the email server (POP/IMAP server) replaces the display name with the one defined for an email account.
Technically, this entry will be included in the *From:* field in the body of the notification email.

MailServerFromAddress defines the from address of the sender, which is the sender's email address presented in an email client.
Note that, for authenticated email transmissions, there is a chance that the email server (POP3/IMAP server) replaces the address with the one defined for an email account.
Technically, this entry will be included in the *From:* field in the body of the notification email, enclosed by pointed brackets.

MailServerReplyToName is the name that will be used by an email client, if a reply is made.
Note, If this entry is not specified, then the MailServerFromName should be used by the email client, in case of a reply.
Note also that for notification emails it could be useful to use a name that indicates that no replies should be sent, using a name like: "Do Not Reply".
Technically, this entry will be included in the *Reply-To:* field in the body of the notification email.

MailServerReplyToAddress is the address that will be used ab email client, if a reply is made.
Note, If this entry is not specified, then the MailServerFromAddress should be used by the email client, in case of a reply.
Technically, this entry will be included in the *Reply-To:* field in the body of the notification email, enclosed by pointed brackets.

UseTLS tells whether TLS security should be used for the SMTP communication.
Check with your SMTP configuration if this is required.
Typically TLS is required when port 587 is used, while not available when port 25 is used.
Valid options for this property are:

- None (Don't use TLS)
- TLS_1 (Use TLS version 1)
- TLS_1_2 (Use TLS version 1.2)

ConfigMessage defines an optional text to the appended to the notification emails. The preceeding content of the email will be the history log of the issue. Example: "Do not reply to this message".

2.8.6 Format of swNotificationServer.exe.nlog file

The swServerMonitorService.exe.nlog has the following content:

```
<?xml version="1.0" encoding="utf-8" ?>
<!-- XSD manual extracted from package NLog.Schema:
https://www.nuget.org/packages/NLog.Schema-->
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd" xsi:schemaLocation="NLog NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
       autoReload="true">
      <!--internalLogFile="c:\temp\console-example-internal.log"
      internalLogLevel="Info" >-->

  <!-- the targets to write to -->
  <targets>
    <!-- write logs to file -->
      <target xsi:type="File" name="logfile" fileName="NotificationServer.log"
             layout="${longdate} [${threadid}] - ${level:uppercase=true}
${message}${onexception:${newline}EXCEPTION\: ${exception:format=ToString}}" />

    <target xsi:type="ColoredConsole" name="logconsole" detectConsoleAvailable="true"
             layout="${longdate} [${threadid}] - ${level:uppercase=true}
${message}${onexception:${newline}EXCEPTION\: ${exception:format=ToString}}" />
  </targets>

  <!-- rules to map from logger name to target -->
  <rules>
    <logger name="*" minlevel="Info" writeTo="logfile,logconsole" />
  </rules>
</nlog>
```

In the target with name="logfile", you can update the `fileName=` to include the absolute path to the location in your installation where you want the log to be generated. If no path is provided, the file will generate in the same location as the executable.

2.8.7 Format of .props file

The .props file controls the logging that is done by SystemWeaver applications, mostly the servers although also for the swExplorer client.

The .props files have the following content (with typical values):

```
# Set internal debugging
# log4d.debug=TRUE

log4d.rootCategory=INFO,Fil1
#log4d.rootCategory=DEBUG,Fil1

# Global disable level
# log4d.disable=INFO

# Override global disable level
# log4d.disableOverride=TRUE

# Set category factory
# log4d.categoryFactory=TLogDefaultCategoryFactory

# Set root priority
# log4d.rootCategory=WARN,ODS

# Establish category hierarchy
#log4d.category.myapp=INHERITED,Mem1
#log4d.category.myapp.more=DEBUG,Mem2
#log4d.category.myapp.other=INFO,Mem3,#Email
```

```
#log4d.category.alt=ERROR,Mem4,Fil1

# Myapp.other category doesn't log to its parents
#log4d.additive.myapp.other=FALSE

# Create root appender
#log4d.appender.ODS=TLogODSAppender
#log4d.appender.ODS.layout=TLogSimpleLayout

# Create a file appender
log4d.appender.Fil1=TFastLogFileAppender
log4d.appender.Fil1.append=TRUE
log4d.appender.Fil1.fileName=C:\swServerMonitorService.log
log4d.appender.Fil1.errorHandler=TLogOnlyOnceErrorHandler
log4d.appender.Fil1.layout=TLogPatternLayout
log4d.appender.Fil1.layout.pattern=%d [%t] %-5p %-6c %x - %m%n
log4d.appender.Fil1.layout.dateFormat=yyyy-mm-dd hh:mm:ss,zzz
```

The different entries of this file are according to the Apache log4j logging service for Java (http://logging.apache.org/). The values in the sample file above should normally be retained including the "#" comment character, but the entries below are often customized:

log4d.appender.Fil1.append controls whether the log file should be cleared [=false] after restart of the server application, or if the logging should be appended [=true] to an existing log file content.

log4d.appender.Fil1.fileName defines the file path and name of the log file.

2.8.8 Format of swServerMonitorService.settings file

The .settings file, used by the swServerMonitorService, has the following content:

```
<SystemWeaverMonitor wcfServicePort="">
<Servers>

  <Server restart="true" id="">
    <exe>C:\work\SystemWeaver\bin\swDBServer.exe</exe>
    <ini>C:\work\SystemWeaver\bin\swServer.ini</ini>
    <props>C:\work\SystemWeaver\bin\swServer.props</props>
  </Server>

  <!--Server id="">
    <exe>C:\work\SystemWeaver\bin\swDBServer.exe</exe>
    <ini>C:\work\SystemWeaver\bin\swServer2.ini</ini>
    <props>C:\work\SystemWeaver\bin\swServer2.props</props>
  </Server-->

  <Server restart="true" startDelay="10000" id="">
    <exe>C:\Work\Restapi\SystemWeaver.RestApi.exe</exe>
  </Server>

  <Server restart="true" startDelay="10000" id="">
    <exe>C:\work\SystemWeaver\bin\swNotificationServer.exe</exe>
    <ini>C:\work\SystemWeaver\bin\swNotificationServer.ini</ini>
    <props>C:\work\SystemWeaver\bin\swNotificationServer.props</props>
  </Server>

</Servers>
</SystemWeaverMonitor>
```

The different entries are described below.

<SystemWeaverMonitor> should contain one **<Servers>** tag. It has one required attribute called *wcfServicePort*. Enter a unique port number for the monitor service.

A **<Server>** entry is needed for each SystemWeaver server application to be executed on as a service. It has 3 attributes:

- *id* is required, and is used to identify the application.
- *restart* is optional, and controls automatic restart of the server application, in case it has been terminated for some reason. This option makes the installation more robust to unexpected failures. Possible values are 'true' or 'false'
- *startDelay* is optional, and controls if the server start will be delayed. You can enter a value in number of milliseconds. This is recommended for servers such as swNotificationServer, or any other server application which should start just after the SystemWeaver main server executable is started. Example: startDelay="10000" (for 10 seconds).

The **<exe>**, **<ini>** and **<props>** entries should refer to the installation folder on a local drive. Note that when the server applications are executed as part of a service they do not have any home directory, so using these entries is the only way to direct the files to the applications.

Note that:

- Each server should have its unique .ini and .props file since several settings like the ports of the servers and log files must be unique to each running server application. The same .exe file can however be used for multiple servers (unless different server versions are needed).
- Depending on the Windows Server version there can be limitations in the number of applications (.exe files) run by a service.
- An application started by a service does not have a "home directory". This means that the file paths in the .settings file (and .ini - check) must be absolute, and not relative.
- All file paths may include space characters, with no need to use quotation marks or apostrophes to enclose the path string.
- The *restart* attribute should have the value 'false' (or be missing since 'false' is default) until you have started the service fault-free for the first time, since the automatic restart behavior may make other fault tracing troublesome as the service may start repeatedly when errors are detected.
- The files used by a SystemWeaver server should always be local to the server computer. Using any network file server may affect access performance and reliability.

2.8.9 Format of swServerMonitorService.exe.nlog file

The swServerMonitorService.exe.nlog, used by the swServerMonitorService, has the following content:

```
<?xml version="1.0" encoding="utf-8" ?>
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      autoReload="true"
      throwExceptions="false">
```

```
  <variable name="appName" value="YourAppName" />

  <targets async="true">
    <target xsi:type="File"
             name="default"
             layout="${longdate} - ${level:uppercase=true}: ${message}${onexception:${newline}
EXCEPTION\: ${exception:format=ToString}}"
             fileName="D:\monitor.log"
             keepFileOpen="false"
             />

    <target xsi:type="EventLog"
             name="eventlog"
             source="${appName}"
             layout="${message}${newline}${exception:format=ToString}"/>
  </targets>
  <rules>
    <logger name="*" writeTo="default" minlevel="Info" />
    <logger name="*" writeTo="eventlog" minlevel="Error" />
  </rules>
</nlog>
```

In the target with name="default", update the `fileName=` line to include the absolute path to the location in your installation where you want the monitor.log to be generated.

2.8.10 Format of swServerMonitorWebAppHost.exe.config

The swServerMonitorWebAppHost.exe.config file has the following content:

Example:

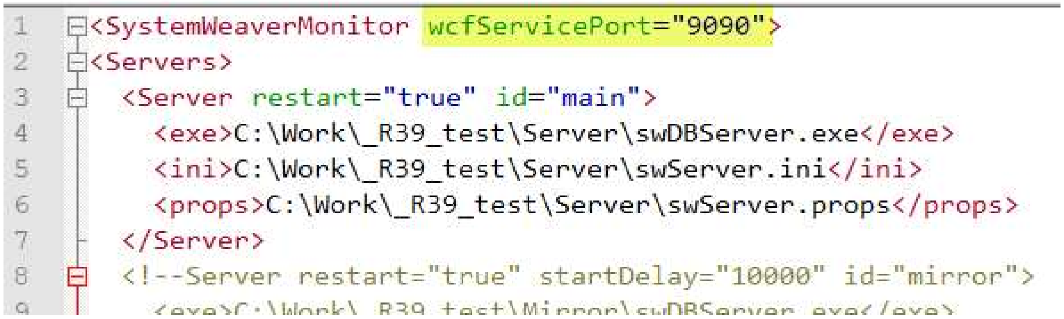
```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
  </startup>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="Microsoft.Owin" publicKeyToken="31bf3856ad364e35"
culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-3.1.0.0" newVersion="3.1.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="Microsoft.Owin.Security" publicKeyToken="31bf3856ad364e35"
culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-3.1.0.0" newVersion="3.1.0.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>

  <appSettings>
    <add key="WebHostPort" value="5000" />
    <add key="WebHostAddress" value="localhost" />
    <add key="ServerMonitorWCFServiceAddress" value="http://localhost:9090/ServerMonitor" />
  </appSettings>

</configuration>
```

The only entries to modify are the two **<add>** entries in <appSettings>.

- key="WebHostPort": for *value*, enter the port to be used by the tool.
- key="WebHostAddress": for *value*, enter the address from which the tool can be accessed.
    - "localhost" means the tool can only be accessed from the server machine.
    - "*" means that the tool can be accessed from any machine
- key="ServerMonitorWCFServiceAddress": for *value*, enter the port used by the *SystemWeaver Monitor Service*. This value should match the value for *wcfServicePort* in the Format of swServerMonitorService.settings file



```
<SystemWeaverMonitor wcfServicePort="9090">
<Servers>
  <Server restart="true" id="main">
    <exe>C:\Work\_R39_test\Server\swDBServer.exe</exe>
    <ini>C:\Work\_R39_test\Server\swServer.ini</ini>
    <props>C:\Work\_R39_test\Server\swServer.props</props>
  </Server>
  <!--Server restart="true" startDelay="10000" id="mirror">
    <exe>C:\Work\_R39_test\Mirror\swDBServer.exe</exe>
```

2.8.11 Format of swExplorerLauncher.conf file

The swExplorerLauncher.conf file, used by the swExplorerLaucher should have the following format:

```
<Servers>
  <Server ip="main" port="443">
    <Alt ip="sys1" port="3000"/>
    <Alt ip="sys2" port="3000"/>
    <Alt ip="213.115.2.233" port="443"/>
    <Alt ip="2001:db8:85a3:0:0:8a2e:370:7334" port="443"/>
  </Server>
  <Server ip="main2" port="1768">
     <Alt ip="sys3" port="3000"/>
  </Server>
</Servers>
```

Each entry under a <Server> tag represents alternative connections or address formats to the same SystemWeaver database.

For convention, the server according to the ip attribute of the <Server> tag should be a main server, with direct connection to the database.

The swExplorerLauncher.conf file should then be located in the same (Client) directory as the swExplorerLauncher.exe

2.8.12 Format of Systemite.SystemWeaver.TcpSubServer.exe.nlog

The Systemite.SystemWeaver.TcpSubServer.exe.nlog, used by the TcpSubServer in a split server architecture, has the following content:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      autoReload="true"
      throwExceptions="false">

  <variable name="appName" value="YourAppName" />

  <targets async="true">
    <target xsi:type="File"
             name="default"
             layout="${longdate} [${threadid}] - ${level:uppercase=true}
${message}${onexception:${newline}EXCEPTION\: ${exception:format=ToString}}"
             fileName="D:\TcpSubServer.log"
             keepFileOpen="false"
             />

    <target xsi:type="EventLog"
             name="eventlog"
             source="${appName}"
             layout="${message}${newline}${exception:format=ToString}"/>

    <!--target name="statisticsLogFileAsync"
            xsi:type="AsyncWrapper"
            overflowAction="Grow"
            queueLimit="10000"
            batchSize="100"
            timeToSleepBetweenBatches="50">
      <target xsi:type="File"
              name="statisticsLogFile"
              fileName="C:\temp\Statistics.log">
        <layout xsi:type="JsonLayout">
          <attribute name="time" layout="${longdate}" />
          <attribute name='seqId' layout='${event-properties:seqId}' />
          <attribute name='sessionId' layout='${event-properties:sessionId}' />
          <attribute name='ip' layout='${event-properties:ip}' />
          <attribute name='userId' layout='${event-properties:userId}' />
          <attribute name='userName' layout='${event-properties:userName}' />
          <attribute name='email' layout='${event-properties:email}' />
          <attribute name='appName' layout='${event-properties:appName}' />
          <attribute name='event' encode='false'>
            <layout type='JsonLayout' renderEmptyObject="false">
              <attribute name='type' layout='${event-properties:eventType}' />
            </layout>
          </attribute>
          <attribute name='call' encode='false'>
            <layout type='JsonLayout' renderEmptyObject="false">
              <attribute name='type' layout='${event-properties:callType}' />
            </layout>
          </attribute>
          <attribute name='action' encode='false'>
            <layout type='JsonLayout' renderEmptyObject="false">
              <attribute name='type' layout='${event-properties:actionType}' />
            </layout>
          </attribute>
        </layout>
      </target>
    </target-->
  </targets>

  <rules>
    <logger name="*" writeTo="default" minlevel="Info" />
    <logger name="*" writeTo="eventlog" minlevel="Error" />
    <!--logger name="StatisticsLogger" minlevel="Info" writeTo="statisticsLogFileAsync" /-->
  </rules>
</nlog>
```

In the first target with name="default", update the `fileName=` line to indicate the TCPSubServer.log file name and the correct path to it in your installation. This will differentiate it from the swServer.log and ensure two separate log files.

Statistics Log
The configuration file also offers an optional logging of certain user operations and usage statistics. This is the target named="statisticsLogFile". The log is JSON formated and can easily be read, analyzed and visualized using an external tool of choice, such as ELK (Elasticsearch, Logstash, and Kibana). Note that as with any logging, sufficient disk space must be available for storing the files. To obtain the output, activate the `target name="statisticsLogFileAsync"` (it is commented out in the above example). `Then,` update the `fileName=` line to indicate the file name and the correct path to it in your installation. The default filename is Statistics.log. Lastly, activate the `logger name="StatisticsLogger"` rule (it is commented out in the example).

To find out more about the Statistics log see Working with the statistics log.

Tip: NLog enables you to divide the TcpSubServer log files into smaller volumes if desired. (https://nlog-project.org/)

2.8.12.1 Working with the statistics log

The Systemite.SystemWeaver.TcpSubServer.exe.nlog offers a configurable option for logging specific user operations and usage statistics. The actions, events and calls currently included in the log are listed below. The events and calls from both the client application and the API will be included, whereas only actions performed in the client application will be logged. They can prove to be useful information for Administrators.

User Activity Statistics
The following operations in the server are included in the log file.

Actions

- PrintDocument - The printing of a Document or Report.
- ExportToXML - The export of XML files, i.e., XML generation and XML Export.
- ExportToExcel - The export of data to MS Excel (or csv), i.e., export data from a configured grid.
- SaveToRTF - The saving of a Document or Report to an RTF and RVF file save a SysW document or report to RTF
- ItemViewLoaded - One of the following views has been loaded. The name of the previously loaded view is also provided, which offers an easy way to get the user flow.
  - Configure the explorer views, i.e., Coverage, Part list editor
  - Configurable views - config item, i.e., Chart, Graph, Grid, Report
  - Extension views, e.g., HARA
  - Project view- issues, i.e., public views

Calls
The following deletion calls are logged:

- deleteItem - An item is deleted from the server.
- deletePart - A part is deleted from the server.
- deleteIssue - An issue is deleted from the server.

The following user administration calls are logged:

- CreateUser - A user account is created.
- UserActivation - A user account is activated or deactivated.
- DeleteUser - A user account is deleted.

Events

Usage Statistics

The following usage events are logged:

- Login - A client login to the server.
- LoginCount
  - A count of the number of user logins (i.e., active connections)
  - A count of the number of concurrent Read/Write licenses in use
  - A count of the number of concurrent Viewer licenses in use

View Statistics

View events are logged when users load the following views.

- Configure the explorer views, i.e., Coverage, Part list editor
- Configurable views - config item, i.e., Chart, Graph, Grid, Report
- Extensions, e.g., HARA
- Project view- issues, i.e., public views

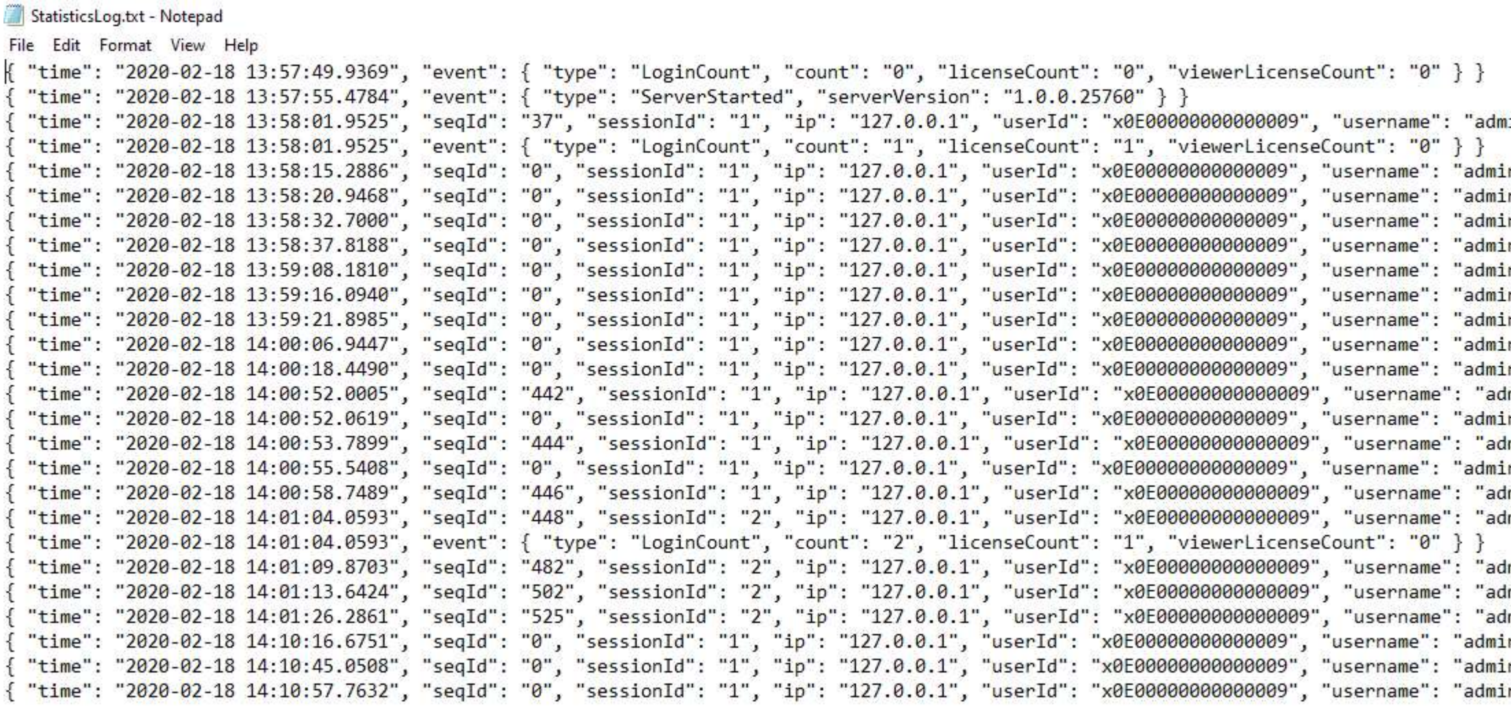
Server Start

Each server start is logged along with server version #.

Reading and Analyzing the Log

The Statistics log is JSON formated and can easily be read, analyzed and visualized using an external tool of choice, such as ELK (Elasticsearch, Logstash, and Kibana) which is open source. One solution is to install, for example, docker and include an image of ELK there.

Example:



```
{ "time": "2020-02-18 13:57:49.9369", "event": { "type": "LoginCount", "count": "0", "licenseCount": "0", "viewerLicenseCount": "0" } }
{ "time": "2020-02-18 13:57:55.4784", "event": { "type": "ServerStarted", "serverVersion": "1.0.0.25760" } }
{ "time": "2020-02-18 13:58:01.9525", "seqId": "37", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "adm
{ "time": "2020-02-18 13:58:01.9525", "event": { "type": "LoginCount", "count": "1", "licenseCount": "1", "viewerLicenseCount": "0" } }
{ "time": "2020-02-18 13:58:15.2886", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
{ "time": "2020-02-18 13:58:20.9468", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
{ "time": "2020-02-18 13:58:32.7000", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
{ "time": "2020-02-18 13:58:37.8188", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
{ "time": "2020-02-18 13:59:08.1810", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
{ "time": "2020-02-18 13:59:16.0940", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
{ "time": "2020-02-18 13:59:21.8985", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
{ "time": "2020-02-18 14:00:06.9447", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
{ "time": "2020-02-18 14:00:18.4490", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
{ "time": "2020-02-18 14:00:52.0005", "seqId": "442", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "ad
{ "time": "2020-02-18 14:00:52.0619", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
{ "time": "2020-02-18 14:00:53.7899", "seqId": "444", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "ad
{ "time": "2020-02-18 14:00:55.5408", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
{ "time": "2020-02-18 14:00:58.7489", "seqId": "446", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "ad
{ "time": "2020-02-18 14:01:04.0593", "seqId": "448", "sessionId": "2", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "ad
{ "time": "2020-02-18 14:01:04.0593", "event": { "type": "LoginCount", "count": "2", "licenseCount": "1", "viewerLicenseCount": "0" } }
{ "time": "2020-02-18 14:01:09.8703", "seqId": "482", "sessionId": "2", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "ad
{ "time": "2020-02-18 14:01:13.6424", "seqId": "502", "sessionId": "2", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "ad
{ "time": "2020-02-18 14:01:26.2861", "seqId": "525", "sessionId": "2", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "ad
{ "time": "2020-02-18 14:10:16.6751", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
{ "time": "2020-02-18 14:10:45.0508", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
{ "time": "2020-02-18 14:10:57.7632", "seqId": "0", "sessionId": "1", "ip": "127.0.0.1", "userId": "x0E00000000000009", "username": "admi
```

Recommendations

The shipper application used, e.g., beats, should be installed on the same server machine as the SystemWeaver server application and Statistics log file

The docker should be installed on a different server than where SystemWeaver server is installed

2.8.13 Format of SystemWeaver.RestApi.exe appsettings.json

The SystemWeaver REST API Server is configured using the appsettings.json file.

```
{
  "SystemWeaver": {
    "restApiBaseURL": "localhost",
    "restApiPort": 9000,
    "requireHttps": false,
    "certificateSubjectName": "localhost",
    "systemWeaverHost": "localhost",
    "systemWeaverPort": 8888
    "allowHttp2": false
  },
  "Tokens": {
    "Key": "0123456789ABCDEFX",
    "Issuer": "http://systemite.se",
    "Audience": "http://systemite.se"
  },
  "NLog": {
    "targets": {
      "console": {
        "type": "ColoredConsole",
        "detectConsoleAvailable": true,
        "layout": "${longdate} ${message} ${exception:format=tostring}"
      },
      "file": {
        "type": "File",
        "fileName": "${basedir:fixtempdir=true}/SystemWeaver.RestApi.log",
        "layout": "${longdate} ${message} ${exception:format=tostring}"
      },
      "requestConsole": {
        "type": "ColoredConsole",
        "detectConsoleAvailable":  true,
        "layout": "${longdate} ${message}"
      },
      "requestFile": {
        "type": "File",
        "fileName": "${basedir:fixtempdir=true}/SystemWeaver.RestApi.Request.log",
        "layout": "${longdate} ${message}"
      },
      "responseFile": {
        "type": "File",
        "fileName": "${basedir:fixtempdir=true}/SystemWeaver.RestApi.Response${shortdate}.log",
        "layout": {
          "type": "JsonLayout",
          "Attributes": [
            {
              "name": "time",
              "layout": "${longdate}"
            },
            {
              "name": "method",
              "layout": "${event-properties:method}"
            },
            {
              "name": "absolutePath",
              "layout": "${event-properties:absolutePath}"
            },
            {
              "name": "queryString",
              "layout": "${event-properties:queryString}"
            },
            {
              "name": "callDuration",
              "layout": "${event-properties:callDuration}"
            },
            {
              "name": "httpStatus",
              "layout": "${event-properties:httpStatus}"
            },
            {
              "name": "callerIp",
              "layout": "${event-properties:callerIp}"
            },
            {
              "name": "httpUserAgent",
              "layout": "${event-properties:httpUserAgent}"
            },
            {
              "name": "username",
              "layout": "${event-properties:username}"
            },
            {
              "name": "userId",
              "layout": "${event-properties:userId}"
            },
            {
              "name": "apiVersion",
              "layout": "${event-properties:apiVersion}"
            }
          ]
        }
      },
      "startupFile": {
        "type": "File",
        "fileName": "${basedir:fixtempdir=true}/SystemWeaver.RestApi.Response${shortdate}.log",
        "layout": {
          "type": "JsonLayout",
          "Attributes": [
            {
              "name": "time",
              "layout": "${longdate}"
            },
            {
              "name": "restApiPrefix",
              "layout": "${event-properties:restApiPrefix}"
            },
            {
              "name": "restApiHost",
              "layout": "${event-properties:restApiHost}"
            },
            {
              "name": "restApiPort",
              "layout": "${event-properties:restApiPort}"
            },
            {
              "name": "baseUri",
              "layout": "${event-properties:baseUri}"
            },
            {
              "name": "systemWeaverHost",
              "layout": "${event-properties:systemWeaverHost}"
            },
            {
              "name": "systemWeaverPort",
              "layout": "${event-properties:systemWeaverPort}"
            },
            {
              "name": "assemblyFileVersion",
              "layout": "${event-properties:assemblyFileVersion}"
            },
            {
              "name": "assemblyVersion",
              "layout": "${event-properties:assemblyVersion}"
            }
          ]
        }
      }
    },
    "rules": [
      {
        "logger": "SystemWeaver.RestApi.WebService.Setup.RequestLogger",
        "minlevel": "Debug",
        "writeTo": "requestFile,requestConsole",
        "final": true
      },
      {
        "logger": "StartupLogger",
        "minlevel": "Debug",
        "writeTo": "startupFile",
        "final": true
      },
      {
        "logger": "ResponseLogger",
        "minlevel": "Debug",
        "writeTo": "responseFile",
        "final": true
      },
      {
        "logger": "*",
        "minlevel": "Debug",
        "writeTo": "file,console"
      }
    ]
  }
}
```

The "SystemWeaver" section contains the configuration information required for the SystemWeaver REST API server to be set up properly. Below is an excerpt of the configuration file:

```
{
  "SystemWeaver": {
    "restApiBaseURL": "localhost",
    "restApiPort": 9000,
    "requireHttps": false,
    "certificateSubjectName": "localhost",
    "systemWeaverHost": "localhost",
    "systemWeaverPort": 8888
  },
```

The various settings are explained below:

RestApiBaseUrl: The base url of the SystemWeaver REST API Server that REST Clients will connect to.
RestApiPort: The port of the SystemWeaver REST API Server that REST Clients will connect to.
RequireHttps: Indicates whether to use HTTPS or not.
certificateSubjectName: The Subject of the certificate.

Important Note: For production use, HTTPS is strongly recommended to secure the transfer of user credentials between the clients and the server, otherwise the user name and password will be transferred in plain text. Please note that this requires a Certificate.

SystemWeaverHost: The URL of the SystemWeaver server or mirror server that the REST API Server utilizes.
SystemWeaverPort: The port of the SystemWeaver server or mirror server that the REST API Server utilizes.

Configuring Logging

The SystemWeaver REST API uses NLog for logging. There are two main logs produced:

- Request log
- Application log

The Request log logs the API requests from the clients together with their parameters.
The Application log is a standard application log that logs user logins, server communication, exceptions, etc.

In the configuration file, there is a section called "NLog". This section contains the configuration information required for the SystemWeaver REST API server logging to be set up properly. Below is an excerpt of the configuration file:

Each target in the configuration represents a different log output.

Console Logging

The first target with the name "console" outputs logging information into the application's console window.

```
"console": {
  "type": "ColoredConsole",
  "detectConsoleAvailable": true,
  "layout": "${longdate} ${message} ${exception:format=tostring}"
},
```

This target is not intended to be changed.

Application Log

The second target with the name "file" outputs the Application log. By default, this log is stored in the same directory as the SystemWeaver REST API executable with the file name "SystemWeaver.RestAPi.log", as defined in the fileName attribute.

```
"file": {
  "type": "File",
  "fileName": "${basedir:fixtempdir=true}/SystemWeaver.RestApi.log",
  "layout": "${longdate} ${message} ${exception:format=tostring}"
},
```

To change the location of the log file, choose another destination and enter that value in the fileName attribute together with the log filename. The below example shows a configuration that saves the log file to "C:/Logs/SystemWeaver.RestApi.log".

```
  "fileName": "C:/Logs/SystemWeaver.RestApi.log",
```

Request Log

The third and forth targets with the names "requestConsole" and "requestFile" outputs the Request log.

```
"requestConsole": {
  "type": "ColoredConsole",
  "detectConsoleAvailable":  true,
  "layout": "${longdate} ${message}"
},

"requestFile": {
  "type": "File",
  "fileName": "${basedir:fixtempdir=true}/SystemWeaver.RestApi.Request.log",
  "layout": "${longdate} ${message}"
},
```

By default, the requestFile log is stored in the same directory as the SystemWeaver REST API executable with the file name "SystemWeaver.RestAPi.Request.log", as defined in the fileName attribute. To change the location of the log file, choose another destination and file name and enter that value in the fileName attribute, in the same way as shown for the Application log example above.

Configuring the Logging Rules

There are two logging rules defined for the SystemWeaver REST API Server, one that applies to the Request logging, name= "SystemWeaver.RestApi.WebService.Setup.RequestLogger" and one that applies to Application logging, name="*". The logger rules are used to define the lowest log levels that are to be logged. For more information about log levels, see: NLog Log Levels.

```
    "rules": [
      {
        "logger": "SystemWeaver.RestApi.WebService.Setup.RequestLogger",
        "minlevel": "Debug",
        "writeTo": "requestFile,requestConsole",
        "final": true
      },
...
      {
        "logger": "*",
        "minlevel": "Debug",
        "writeTo": "file,console"
      }
    ]
```

Request Log Rule
The Request log only outputs information on log level "Info". To disable it, set minLevel to "Off" or "Warning".

Application Log Rule
For the Application Log rule "*", it is recommended to keep the default settings.

2.9 Database and Server Set Up

The following paragraphs describe the actions and alternatives required for the initial set up of a SystemWeaver server and database when SystemWeaver is deployed in an organization.

2.9.1 The Empty database case

SystemWeaver is normally supplied with an "empty" SystemWeaver database including the two accounts "root", with Root rights, and "admin" with Administrator rights. Passwords to these accounts are also included in the delivery.

Recommendations for the use of these accounts can be found in [2] SystemWeaver Access Rights and Administration.

The "empty" database is not entirely empty since the meta model is populated with some standard item types. Additional item types and other settings can be defined using the options described below.

The swDatabaseManager also includes support for the creation of empty databases, refer to the section about the swDatabaseManager.

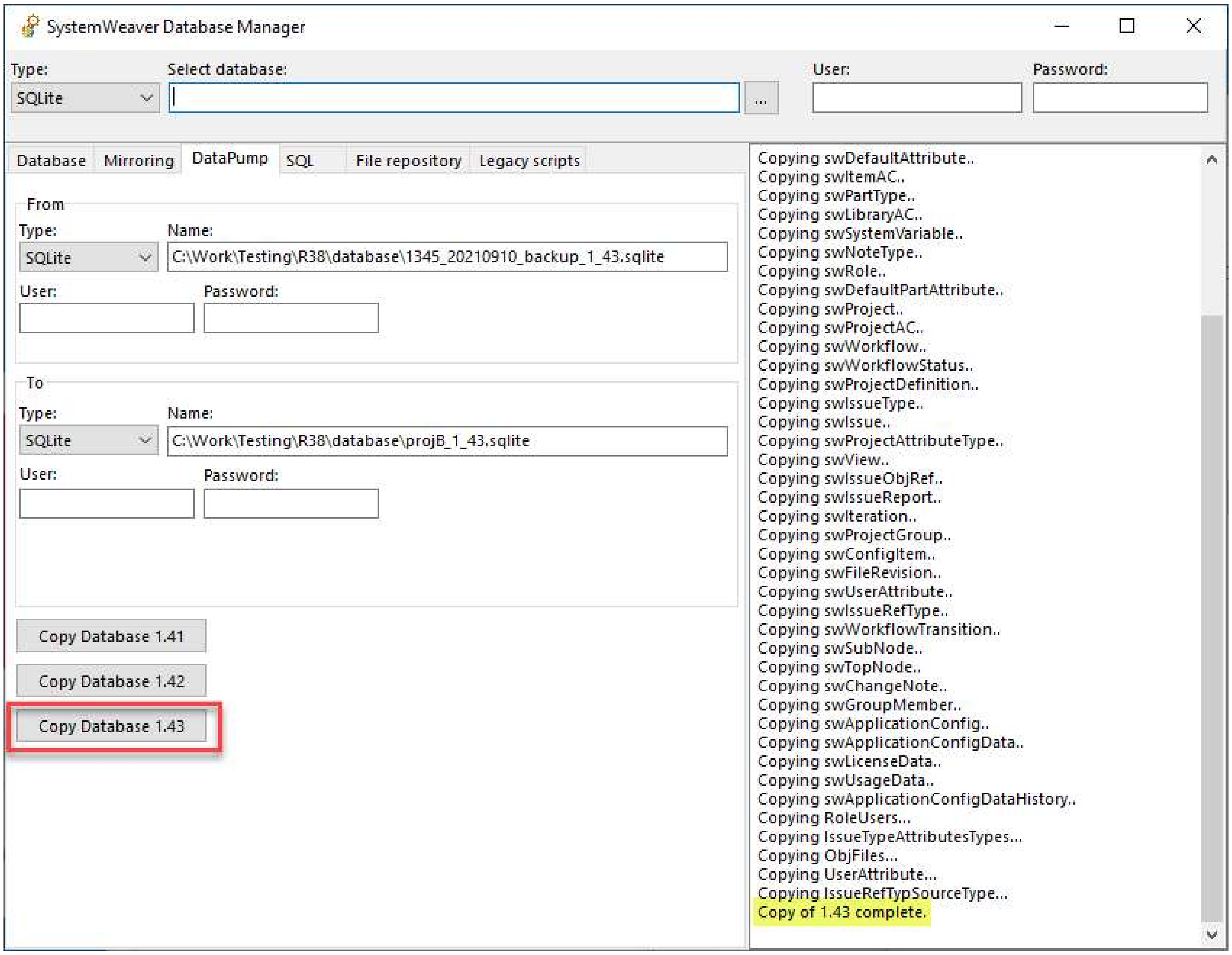
SystemWeaver is normally also supplied with an empty file repository database file. This file, typically called "EmptyFileRepository.sqlite" should be renamed to a suitable name, and located for example in the same folder as the SystemWeaver database. The file server entry in the swServer.ini-file (see Format of swServer.ini file) should be set to the file path.
The swDatabase Manager can also be used for the creation of empty file repository database files or folders.

2.9.2 Cloning a database

There are some cases where an independent copy of an existing database is needed. Mostly this is for testing new releases of SystemWeaver releases before deployment, or for testing new meta models.

When SystemWeaver uses a single file database, like an SQLite database it is possible to make a copy of the entire database file using Windows file copying operations. The Copies should preferably be made from back-up copies of a database, to assure the integrity of the copy.

Any SystemWeaver database can also be replicated into another database using "data pump" feature of the SystemWeaver Database Manager application as shown below.



Type: SQLite
Select database:
User:
Password:

From
Type: SQLite
Name: C:\Work\Testing\R38\database\1345_20210910_backup_1_43.sqlite
User:
Password:

To
Type: SQLite
Name: C:\Work\Testing\R38\database\projB_1_43.sqlite
User:
Password:

```
Copying swDefaultAttribute..
Copying swItemAC..
Copying swPartType..
Copying swLibraryAC..
Copying swSystemVariable..
Copying swNoteType..
Copying swRole..
Copying swDefaultPartAttribute..
Copying swProject..
Copying swProjectAC..
Copying swWorkflow..
Copying swWorkflowStatus..
Copying swProjectDefinition..
Copying swIssueType..
Copying swIssue..
Copying swProjectAttributeType..
Copying swView..
Copying swIssueObjRef..
Copying swIssueReport..
Copying swIteration..
Copying swProjectGroup..
Copying swConfigItem..
Copying swFileRevision..
Copying swUserAttribute..
Copying swIssueRefType..
Copying swWorkflowTransition..
Copying swSubNode..
Copying swTopNode..
Copying swChangeNote..
Copying swGroupMember..
Copying swApplicationConfig..
Copying swApplicationConfigData..
Copying swLicenseData..
Copying swUsageData..
Copying swApplicationConfigDataHistory..
Copying RoleUsers...
Copying IssueTypeAttributesTypes...
Copying ObjFiles...
Copying UserAttribute...
Copying IssueRefTypSourceType...
Copy of 1.43 complete.
```

2.9.3 Import of Meta Model

The Meta Model of a server can be imported from a SystemWeaver file using the Import XML option in the swArchitect application, see *Figure - Import of Meta Model* below.

The XML file could either be a pure meta model file exported from the swArchitect application, or it could be a mixed meta model/model file generated from an item in the swExplorer client.

The import requires SW Architect rights. Refer to [2] SystemWeaver Access Rights and Administration for a description of the root right.

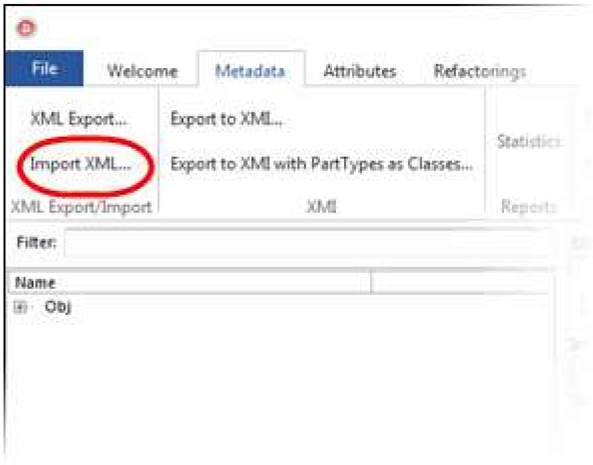
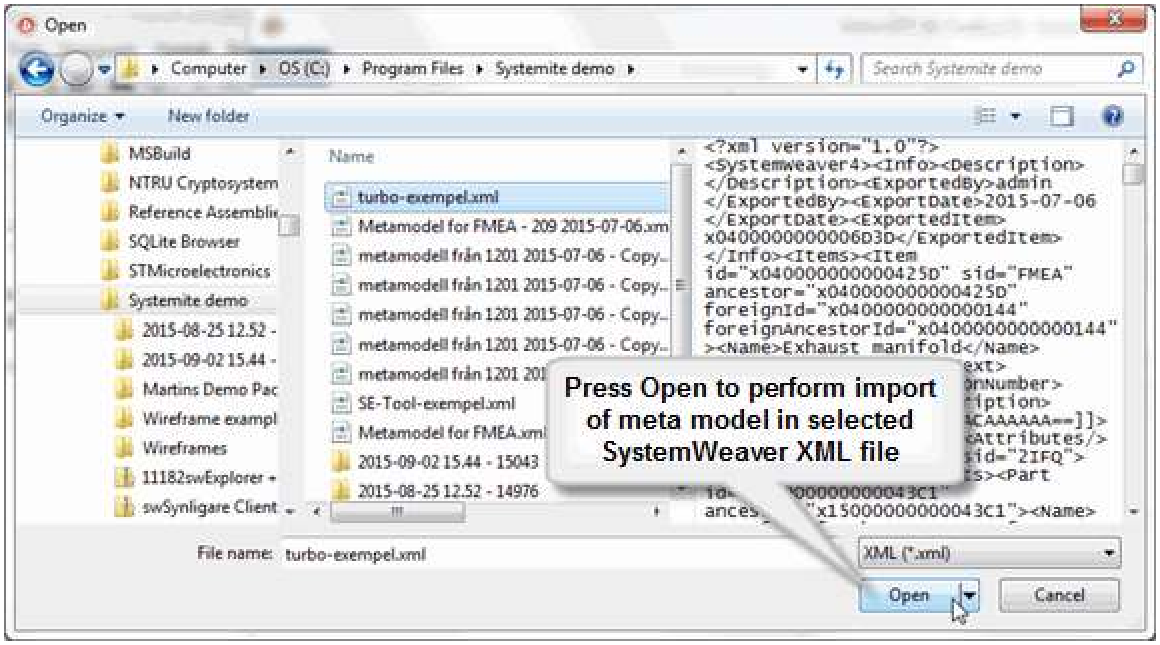


Figure - Import of Meta Model

When the **Import XML...** button is pressed, the next step is to select a SystemWeaver XML file:



```
<?xml version="1.0"?>
<Systemweaver4><Info><Description>
</Description><ExportedBy>admin
</ExportedBy><ExportDate>2015-07-06
</ExportDate><ExportedItem>
x0400000000006D3D</ExportedItem>
</Info><Items><Item
id="x040000000000425D" sid="FMEA"
ancestor="x040000000000425D"
foreignId="x0400000000000144"
foreignAncestorId="x0400000000000144"
><Name>Exhaust manifold</Name>
                                      ext>
                                      onNumber>
                                      iption>
                                      CAAAAAA==]]>
                                      Attributes/>
                                      id="2IFQ">
                                      s><Part
id                  00000000043C1"
ances        "x1500000000043C1"><Name>
```

Press Open to perform import of meta model in selected SystemWeaver XML file

The import requires that the file is a SystemWeaver XML file and that there are no conflicts in the SIDs used in the source and target database, otherwise the import completes in an instant when the Open button is pressed.

2.9.4 IP Address and Ports

The *IP Address* of a SystemWeaver Server host is managed by the operating system and should normally* not be set up in the SystemWeaver Server configuration.

The *port number* for SystemWeaver has not been standardized. The Dynamic and/or private ports in the 49152-65535 range are allowed for use by the Internet Assigned Numbers Authority (IANA) but these may be in momentary conflict with other applications depending on the configuration of other applications on the server computer. Refer to e.g. http://www.iana.org/ for the use of port numbers.

In case there is no web server active on the SystemWeaver host computer, then the 443 port assigned to the https protocol can be used with no side effects for the SystemWeaver. In case a web browser tries to connect to the port, the request is simply ignored by the SystemWeaver server.

Since the communication on 443 is assumed to be encrypted, no other network components should assume anything about the content of the communication. Note however that IANA does not approve the use of this number, likely due to the possibility of conflicts with any web server active on the host.

Note that port names must be unique on the host computer, and not used by any other application, including other SystemWeaver servers.

In case the access to the server from the user network will be through a proxy, the SystemWeaver client to server communication set up in SystemWeaver clients can be directed though the standard port (#8080) of the proxy. For access through firewalls and proxies, the server port 443 (https, see above) is useful since this port may offer remote access to SystemWeaver without the need to open new ports on the firewall.

* The IP entry can be used for the case that a server has multiple IP addresses, e.g. is connected to multiple networks. In that case the acquired IP address must however be static/fixed and the IP entry must be set correctly, as the behavior is otherwise undefined; symptoms when the wrong address is used may be confusing e.g. that the server experiences the connection to the client but not the other way around.

3 SystemWeaver applications and utilities

3.1 swDatabaseManager

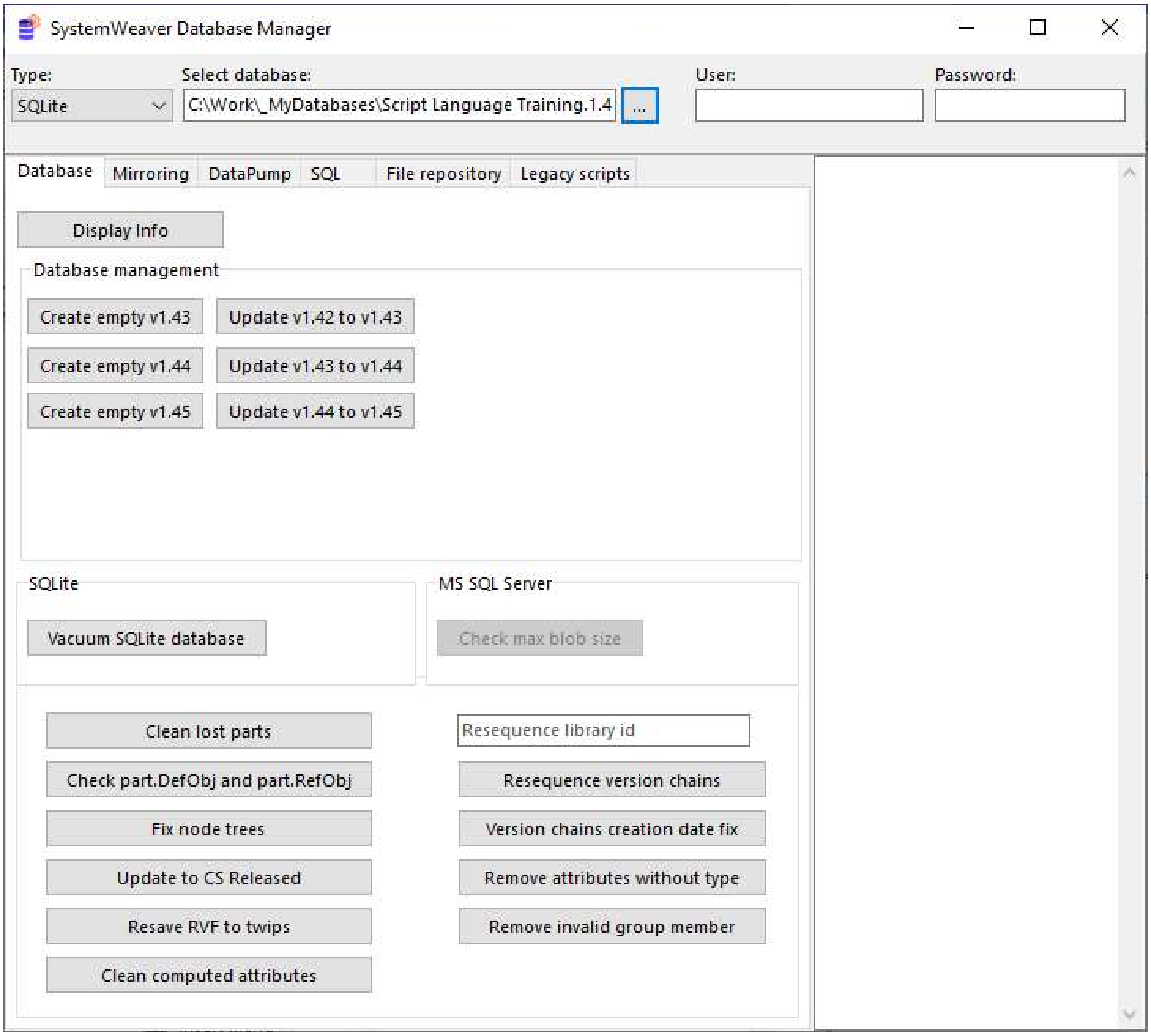
The *swDatabaseManager* is used for incremental upgrades and maintenance of the internal database used by the SystemWeaver server. The database is not visible to any user, but upgrades may be necessary for new releases of SystemWeaver servers. If required, upgrade instructions will include how to perform the specific upgrades for the release using the swDatabaseManager.

Note: The swDatabaseManager application is not for general distribution and should only be used by an authorized system administrator.

The swDatabaseManager operates directly against the database.

When the swDatabaseManager is used, it requires sole access to the database. No other SystemWeaver servers or clients should be connected to the database at the same time, since data may get corrupt.
Stop the database and restart again only after the upgrade is performed.

Since an upgrade will require a new server version, the server application should normally also be closed*.



Type: SQLite

Select database: C:\Work_MyDatabases\Script Language Training.1.4

Figure - swDatabaseManager

As seen in the figure above, the swDatabaseManager Database tab includes support for displaying database version, creation of empty databases for a specific version, upgrades of databases, and specific repairs of databases. (The DataPump tab is described in the section Cloning a database.)

Empty databases are created in the following way for file-based databases like SQLite:

1. Enter file path and name of desired database in the Select database field. File extension should be .sqlite for SQLite**.
2. Press the appropriate Create Empty vX.XX button. The file will be generated in the path location.

The window to the right is a log window that will display the result of all performed operations. In the example above, the log window displays the result of the Display Info operation.

* You cannot replace an EXE file when it is running, and you should not change the name of the application file.
** Any extension works for SQLite, but this extension makes it easier to identify SQLite database files.

3.1.1 File repository

Purpose

This is where you manage where and how the SystemWeaver File repository is stored on disk. You can create new empty file repositories to be used by a server installation or change the file repository type.

Background

The file repository can be one of two possible formats, database or file-based.

DB

The file repository is stored in one big file. This is the easiest representation to handle.

Files

The file repository is stored using the file system in a custom file structure. When the file repository grows it is a good idea to move to this format since it works with backup.

Usage

First there are 2 buttons which allows you to create new empty file repositories:

Create file repository (DB)

Press this to create a database based file repository. You will be asked to specify a filename, the name of the database in which the file repository will be stored. The default extension is .sqlite.

Create file repository (Files)

Press this to create a file repository of type DB. You will be asked to specify a filename, the name of the directory which will contain the file repository structure. Please select a new directory and not an existing one. A directory will be created and an initial structure.

Note: You must never manually edit this structure or it's files. This may corrupt the file store. You can move the file store to another location by copying the entire directory including all it's subdirectories and files.

Convert from database to filebased

If you are using a database based file repository and want to switch to using a file-based file repository you can do this in the "Convert from database to filebased" section:

First you have to select the original database based file repository be clicking "Select database". You will be prompted to select the database.

Then you have to specify where the new file-based repository should be located by clicking "Select directory". You will be prompted to select the location of, and name of, the new directory where you want the repository copy to be placed.

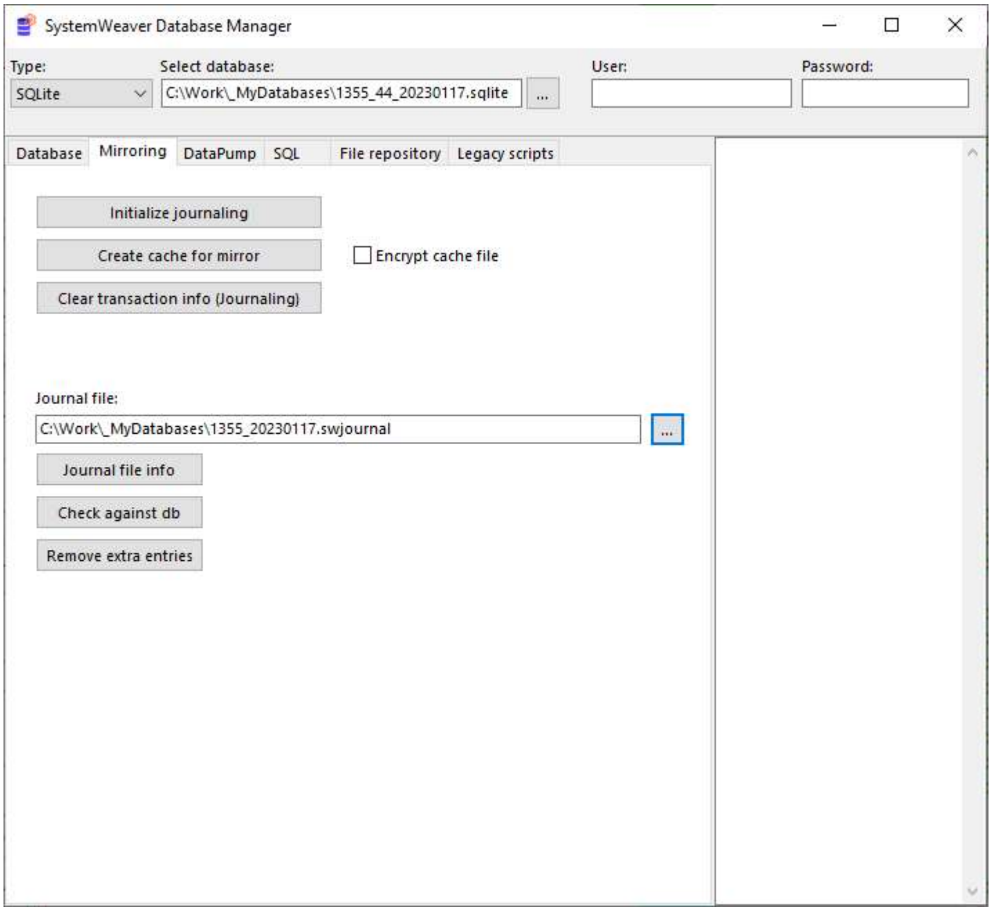
After you have specified the old and new database locations respectively, you can click on Convert which will start the process of copying the files from the original repository to the new repository. This may take a while.

Note. It is good to create a copy of the original repository as an extra backup that you easily can revert to if something unexpected should happen.

3.1.2 Mirroring

Purpose

This is where you manage the files required by the main and mirror server(s) when mirroring is to be used in an installation.



Type: SQLite

Select database: C:\Work_MyDatabases\1355_44_20230117.sqlite

User:

Password:

□ Encrypt cache file

Journal file: C:\Work_MyDatabases\1355_20230117.swjournal

Creating a Journal File

To be able to run a mirror server in an installation, a journal file must be created for the database. If running against an sqlite database, the SystemWeaver server must be stopped, i.e., you must have sole access to the database during file initialization. If running a MS SQL database, there is no need to disconnect the database.

1. Select the database **Type** and **Select database**.
2. Click **Initialize journaling**.
3. Navigate to the desired location for the journal file and give the .swjournal file a name. Consider naming the swjournal file in a way that makes it easy to know which database it belongs to. Click Save to save the new file. The log window will indicate when the journal initialization is completed.

Creating a Cache File

To be able to run a mirror server in an installation, a cache file must be created for each mirror server as this is where the users who are connected to a mirror server obtain their data. The SystemWeaver main server must be stopped, i.e., you must have sole access to the database during file initialization. Alternatively, after activating journaling on the main server, you can take a backup of the database while the SystemWeaver server is up. Then, create the cache file off of the backup so that you do not need to stop the main server.

1. Select the database **Type** and **Select database**.
2. For an encrypted cache file, check **Encrypt cache file**.*
3. Click **Create cache for mirror**.
4. The cache file will need to be saved on the mirror server machine. Navigate to the desired location for the cache file and give the file a name. Consider naming the cache file in a way that makes it easy to know which database and mirror server that it belongs to. Click **Save** to save the new file. The log window will indicate when the cache file creation is completed.
5. Repeat the above steps for each mirror server installation.

Clear transaction info (Journaling) is used to remove the use of journaling. Transaction journaling is required when running a mirror server in an installation. However, there may be scenarios when it must be stopped. A couple of example cases are:

- The existing journal file gets inadvertently lost or deleted
- You no longer want to journal transactions for some reason
- You have taken a copy of a production database for a test environment and do not require journaling there.
- In the unlikely event that the database becomes corrupt and you need to restore to a back-up database

If running against an SQLite database, the SystemWeaver server must be stopped, i.e., you must have sole access to the SQLite database file when removing journaling.

You can also clear journaling via a command line argument:

```
swDatabaseManager.exe clearTransactionInfo type="SQLite" name="C:\temp\db.sqlite"
```

This is useful if you are performing an automated nightly copyover to a QA environment, where journaling is not needed.

Journal file info will display the last transaction id in the file.

Check against db is used to check if a journal file (.journal) belongs to the selected database.

Remove extra entries will bring the journal file into sync with a database. In the event that you need or want to revert to an earlier back-up of a database for whatever reason, the journal file associated with the database that is being replaced will obviously have more transactions in it compared to the back-up database. In other words, all transactions in the journal file that were completed after the back-up database was created will not have been run in the back-up database. Therefore, when you put the back-up database in production, you will first need to remove the extra transaction entries from the journal file.

* The availability of the encryption option is restricted due to EU regulations. Please contact us for more information.

3.2 swExplorer

The swExplorer is the standard multi-purpose client of SystemWeaver. The use of swExplorer is described in [1] SystemWeaver Explorer Help Manual. There are some installation options that may be useful from an administrative perspective:

The *swExplorer.html* file (refer to the section about the swExplorer.html File for details).

The *swExplorer.props* file, which, when available, will create a log file for the swExplorer. For further details, refer to the section Format of .props file.

Extensions

The functionality of the swExplorer client can be extended by means of custom Extension DLLs. Such extensions can be provided by Systemite or a 3rd party that has access to the SystemWeaver Client API and developer's license to the Hydra technology.

Extensions are located in the folder named "swExplorerExtensions".
The folder must be located in the same folder as the swExplorer client.

Command line options

Command line parameters that can be used for configuring client installations:

ItemID	The SystemWeaver ItemID of an item that will be opened when the client is started
LoginName	The login name that will be set in the client, in the Login view of the client
ServerName	The name or IP address of the SystemWeaver server that should be used by the client
ServerPort	The port number of the SystemWeaver server
ProxyType	"HTTPS" or "Socks"
ProxyHost	Hostname of the Proxy
ProxyAuthenticationRequired	"True" or "False"
ProxyLogin	Login for the proxy, which is typically a domain user name
ProxyPassword	Password for the Proxy login, which is typically a domain login password
ProxyPort	Port number used for the proxy. Often 80 or 8080 is used for this purpose
AllowNetworkAuthentication	"True" or "False". True (default) means that LoginName will not be used
AutoLogin	"True" or "False". True means that the client will try to login without any command from user

If any of the parameters are missing, they will be replaced by the most recently used value, and if there is no previously used value they will remain empty.

Example:

```
"C:\Program Files\SystemWeaver\swExplorer.exe" ServerName=sys2 ServerPort=2001
```

3.2.1 System Requirements

Operating System
The SystemWeaver Explorer (swExplorer.exe) application can run on all versions of the Microsoft Windows Operating System that are currently supported by Microsoft.

Microsoft .Net framework
The various plugins used in the SystemWeaver Explorer client require that Microsoft .Net framework version 4.6.2 or later is installed. For the swAdmin2 client, Microsoft .Net framework version 4.6.2 or later is also required. Determine which .NET Framework versions are installed

Visual C++ Redistributable for Visual Studio 2017
This is often already installed on Windows machines because many other programs also use it. But if a machine is basically clean with not much installed, it may be missing. If that is the case, you will need to install it:
Download Microsoft Visual C++ Redistributable for Visual Studio 2017.
You can see which versions are installed under the Control Panel > Apps and features.

Linux Environment
Although SystemWeaver is not available for Linux, we have done limited testing and can recommend VirtualBox as an alternative for developers working in Linux who need to access SystemWeaver which can only be run in Windows. VirtualBox is a free, general-purpose full virtualizer for x86 hardware, targeted at server, desktop and embedded use.

- Helpful Links
  Download and Create Windows 10 Virtual Machine
  - Download VirtualBox for Linux Hosts
  - How to create a Windows 10 virtual machine in VirtualBox
  - Licensing: Frequently Asked Questions

- Tips to Improve Performance
  - https://www.techrepublic.com/article/how-to-improve-virtualbox-guest-performance-in-five-steps/
  - http://www.rawinfopages.com/tips/2016/10/top-tips-to-speed-up-windows-and-linux-in-virtualbox/
  - The SystemWeaver client itself requires at least 2 GB RAM so ensure that your virtual machine has more than 2 GB RAM allocated to it. For organizations with larger databases, we recommend at least 4 GB RAM for the SystemWeaver client. Be aware that the default settings for the virtual machine may not fulfill these recommendations and may need to be adjusted depending on how many applications you are running in the virtual machine.

Disclaimer: Systemite makes no warranty, representation, or guarantee regarding the performance or suitability of VirtualBox when used to access SystemWeaver. The application has been subject to limited testing. Users should conduct and complete performance and other testing of VirtualBox used with SystemWeaver. It is the User's responsibility to independently determine suitability of VirtualBox with the use of SystemWeaver.

VDI Environments
Although SystemWeaver can only be run in Windows, it can be utilized in, for example, Citrix as long as the Citrix version is compatible with the Windows version being run (Windows version must be currently supported by Microsoft). Note that to be able to print to PDF from SystemWeaver, VDI users must be provided access to printers.

Memory(RAM)
We recommend that computers running the SystemWeaver client have at least 2 GB RAM. For larger organizations, we recommend at least 4 GB RAM. Note that the swExplorer is a 32-bit application and it will stop processing when the memory usage reaches approximately 3.5GB.

Processor
Any modern-day standard CPU processor should be sufficient.

Disk Storage
There should be sufficient disk space for the client installation, e.g., 200 MB.

Microsoft Office
There are a number of features in the swExplorer client that provide for an export to a Microsoft Office program. For example, most of the tree or grid type of views in the swExplorer client support export to Microsoft Excel. All versions of Microsoft Office that are supported by Microsoft are supported by the SystemWeaver Explorer client.

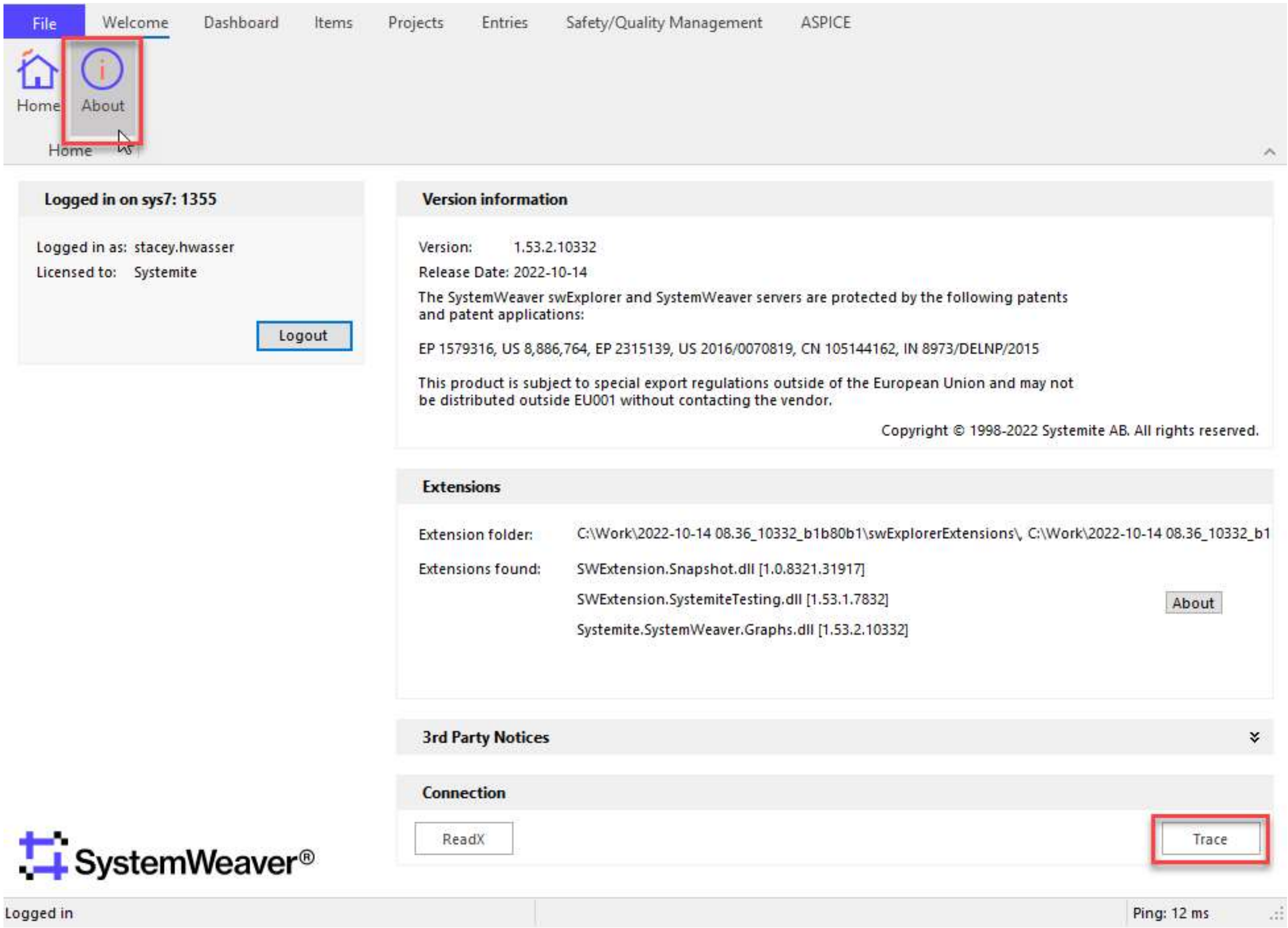
3.2.2 Trace Call

Purpose

The Trace call option displays reading times between client and server to verify latency.

Usage

Navigate to *Welcome* > *About* and click on the **Trace** button.



Logged in on sys7: 1355

Logged in as: stacey.hwasser
Licensed to: Systemite

Version information

Version: 1.53.2.10332
Release Date: 2022-10-14
The SystemWeaver swExplorer and SystemWeaver servers are protected by the following patents and patent applications:

EP 1579316, US 8,886,764, EP 2315139, US 2016/0070819, CN 105144162, IN 8973/DELNP/2015

This product is subject to special export regulations outside of the European Union and may not be distributed outside EU001 without contacting the vendor.

Copyright © 1998-2022 Systemite AB. All rights reserved.

Extensions

Extension folder: C:\Work\2022-10-14 08.36_10332_b1b80b1\swExplorerExtensions\, C:\Work\2022-10-14 08.36_10332_b1

Extensions found: SWExtension.Snapshot.dll [1.0.8321.31917]
SWExtension.SystemiteTesting.dll [1.53.1.7832]
Systemite.SystemWeaver.Graphs.dll [1.53.2.10332]

Connection

Logged in

Ping: 12 ms

The **Trace Result** dialog current displays four reading times:

```
<swClientAPI.TswClientBroker> {
  <TcpSubServer.SWCallProcedure> {
    LockWaitTime: 00:00:00.0000014
    <swServerAPI.TswBroker> {
      <DB> {
        Database: SQLite
        Reading from db
      } 00:00:00.0014947
    } 00:00:00.0015207
  } 00:00:00.0026160
} 00:00:00.0064076
```

The first time (`<DB>`) is the roundtrip time from the TcpSubServer to the database.
The second time (`<swServerAPI.TswBroker>`) is the roundtrip time from the top of the TcpSubServer through the server to the database and back up.
The third time (`<<TcpSubServer.SWCallProcedure>`) is the roundtrip time from the swDBServer through the servers to the database and back up.
The fourth time is the roundtrip time from the Client API through the servers to the database and back up again.

3.3 swExplorerLauncher

The swExplorerLauncher is a utility application that manages references to SystemWeaver objects according to the "swap" URL protocol.

An example of such a URL: "url:swap://209.systemite.net:443/x040000000003ED4D"

If the URL protocol is registered in Windows to use the swExplorerLauncher, any clicking on a swap link will start the swExplorerLauncher and pass the URL as a parameter.

The swExplorerLauncher will then open the URL in the most suitable swExplorer client, connected to a server matching the URL, according to the following steps:

1. The swExplorerLauncher will look at the URL and ask any already existing swExplorer client instances, through a special interface, if they are connected to the SystemWeaver server specified by the URL.
   In this step, it will look up the swExplorerLauncher.conf file, if the file is available, for equivalent server addresses, and use the first matching address.
2. If there is such an instance of the swExplorer, it will ask the client to open the object.
3. If there is no such client, it will start the swExplorer client located in the home folder of the swExplorerLauncher, ask it to connect to the server according the URL, and open the object specified by the URL.
   In this step, it will look up the swExplorerLauncher.conf file (if any), and search through previously used connections (stored in the registry), and use the first matching address.

The swExplorerLauncher uses the command line parameters of the swExplorer in doing this. These are described in the section about the swExplorer.

The "swap" URL protocol needs to be registered on the computer where the benefits of the URL are needed. This can be done with the swExplorerLauncher, when called with a /Register parameter, as done by the *swExplorerLauncherRegister* BAT file:

```
cd /D "%~dp0"
swExplorerLauncher /Register
pause
```

Note that in case an upgrade of the swExplorer is done, in a new folder, a new registration for the new client is required, or else the client that was already registered might be started instead.

The URL protocol can be unregistered in a similar way by using the *swExplorerLauncherUnregister* BAT file.

```
cd /D "%~dp0"
swExplorerLauncher /Unregister
pause
```

3.4 swArchitect

The swArchitect client operates on the meta model of a SystemWeaver database.

The client can also perform refactoring operations, where SystemWeaver models (data) is modified according to a change in the meta model.

The swArchitect should only be used by, and only be distributed to users who have been properly trained in the subject of meta modeling in SystemWeaver.

The swArchitect is further described by the Help manual of the application [3] SystemWeaver Architect Help Manual, distributed along with the application.

3.5 swAdmin

The swAdmin client is used for setting up and administer SystemWeaver user accounts, and roles associated with users.

The swAdmin client also does the following thing:

- Displays currently logged in users
- Sets the security level of the server and other security related tasks
- Registers license keys
- Sets up handling of automatic exception reports

The application is described in ref [4] swAdmin Help Manual.

3.6 swServerCommand

The swServerCommand utility application is used as a scripting interface towards the swServer application, and uses the command line options of to request specific server operations.

The input parameters to the utility, and its capabilities are described in Runtime Backup using SQLite.

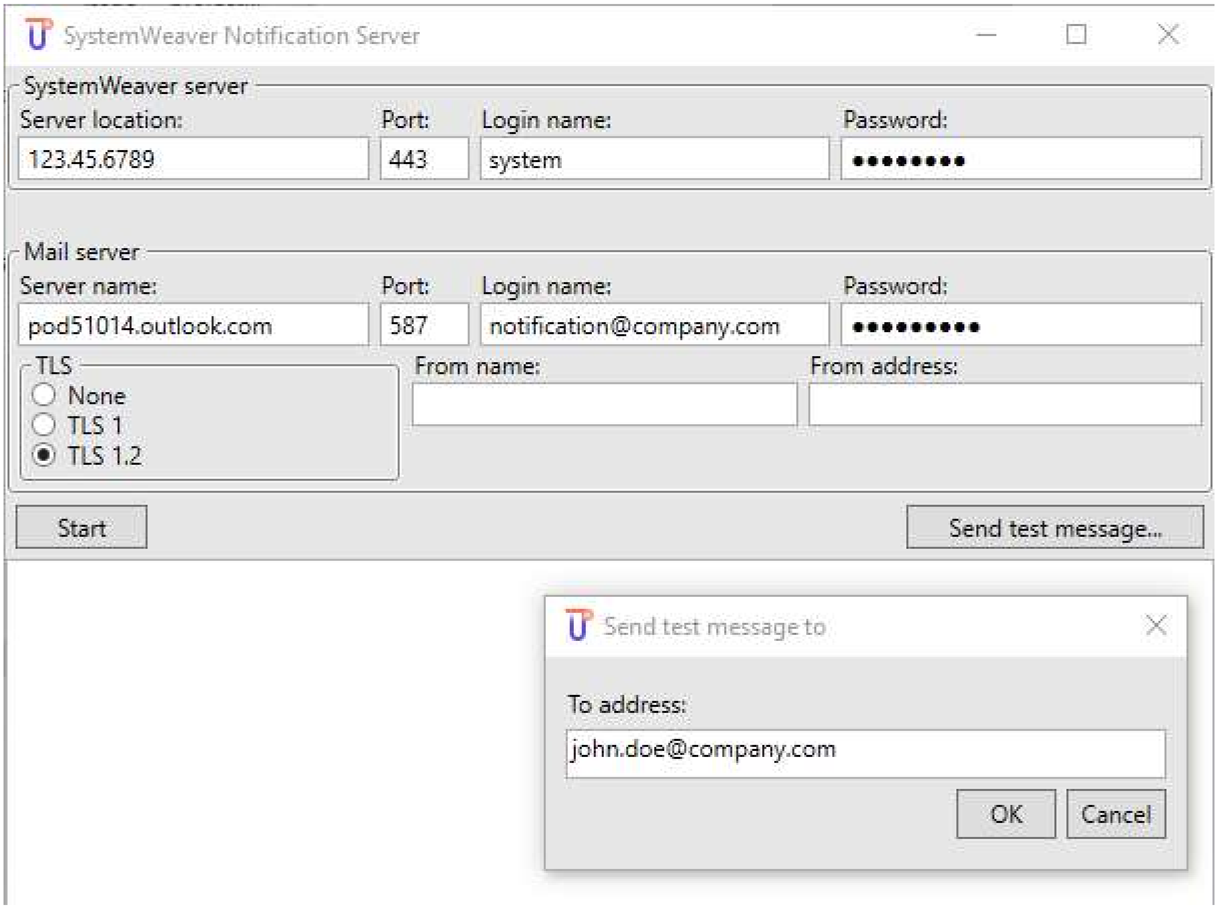
3.7 swTestNotificationServer

The swTestNotificationServer utility application is useful when setting up the configuration of the swNotificationServer. (See SystemWeaver Notification Server for further details). It comes in a 64-bit version only.

The application has an interface where all configuration parameters can be set manually (see table and screenshot below).
Each parameter has a corresponding entry in the INI file of the swNotificationServer (according to Format of swNotificationServer.ini file):

Text box		Entry in INI file
SystemWeaver Server	Server Location	`SystemWeaverServerLocation`
	Port	`SystemWeaverServerPort`
	Login Name	`SystemWeaverLoginName`
	Password	`SystemWeaverPassword`
Email Server	Server Name	`MailServerName`
	Port	`MailServerPort`
	Login Name	`MailServerLoginName`
	Password	`MailServerLoginPassword`
	From Name	`MailServerFromName`
	From Address	`MailServerFromAddress`
	-	`MailServerReplyToName`
	-	`MailServerReplyToAddress`
	Use TLS	`UseTLS`
(hardcoded)		`ConfigMessage`

When all parameters are entered in the text boxes, you can press the **Send Test Message** button to produce a notification email: (the values seen here are only examples)

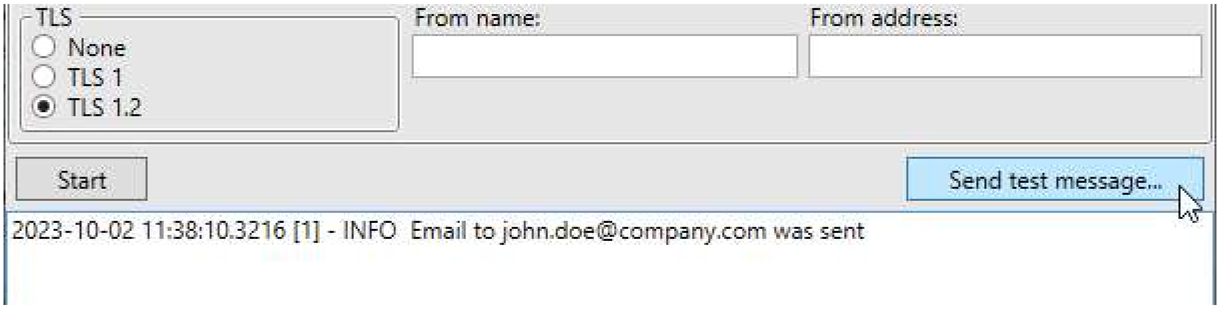


SystemWeaver server
Server location: 123.45.6789
Port: 443
Login name: system
Password: ••••••••

Mail server
Server name: pod51014.outlook.com
Port: 587
Login name: notification@company.com
Password: •••••••••
TLS
○ None
○ TLS 1
● TLS 1.2
From name:
From address:

Send test message to
To address: john.doe@company.com

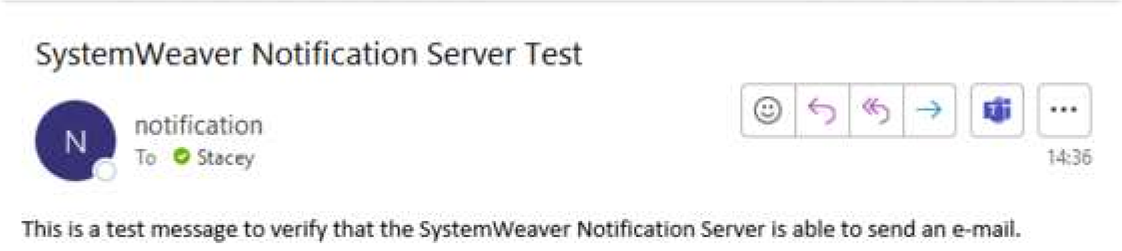
A successful email transfer should return the following status message:



TLS: ○ None ○ TLS 1 ● TLS 1.2

```
2023-10-02 11:38:10.3216 [1] - INFO  Email to john.doe@company.com was sent
```

The default notification message will look like this, when arrived and opened in an email reader:



SystemWeaver Notification Server Test

This is a test message to verify that the SystemWeaver Notification Server is able to send an e-mail.

Note that in this case the email server (MS Outlook) replaced the given *From:* name with the name of the used email account.

Use of Domain Names During Testing
To avoid misdiagnosis during testing, avoid using domain names such as 'company.com','acompany.com', 'user.com' and similar made-up names as these domains do exist and may be identified as spam. The following domain names are reserved by the Internet Engineering Task Force for use in documentation and examples. They are not available for registration. Therefore, they can safely be used:

- example.com
- example.net
- example.org

4 Server operation and maintenance

The goal with SystemWeaver is to offer 24/7 continuous operations. A key to this is the proper configuration and operation of installed servers, as described in the following paragraphs.

Note that although much of the operation and maintenance procedures can be automated, the operation organization must be alert for problems that may occur due to problems in the operating environment. Some of these problems are likely to be detected by users, and the operating organization should therefore have continuous support available at all times when there are users active. An example is when a SystemWeaver server computer is shut down by mistake during operation.

Other problems may not be possible to detect by users until it is too late, and the operating organization should define procedures to avoid such problems. An example is when the storage used for database backups runs out of storage space, which may cause back-up copies to be lost.

4.1 Stopping and starting a SystemWeaver server

The following procedure should always be followed when a SystemWeaver Server Application is stopped and then started again, e.g. after a database backup restoration or upgrade.

1. Make sure that no clients are connected to the server. This can be done by notifying all users in advance by email, telling them to log out before a specific time of day. In case any clients remain connected anyway, e.g. due to the user being absent, the following steps can normally be performed with low risk for loss of data, since most client operations are performed momentarily and atomically. The swAdmin client can list all users currently logged into a server.
2. Make sure that the server application is not busy performing any task. This can be done by looking in the log window of server applications with a GUI, by using the Windows Task Manager or by looking in the log file. Note that some exceptional operations may take excessive time and may cause the server to appear inoperable from the perspective of the clients, when in fact it is busy performing a task that should not be interrupted.
   Also, read the note below*.
3. Stop the SystemWeaver Server Application. See section Stopping a SystemWeaver Server Application for details. Make sure the server has stopped until the next steps are performed. Note that depending on the specific situation, the time for this may vary so the Windows Task Manager should always be used to confirm that the server process has terminated.
4. Make a copy of the original database, the one that was used by the server prior to step 1 above. This is to avoid that the file is overwritten by mistake. Give the copy a suitable name, like appending the file name with current date. For Oracle or SQL Server databases, refer to documentation from Oracle or Microsoft.
5. Make a copy of the log file (typically named "swServer.log"). This is to avoid that the log file is overwritten when the server is restarted, in case the 'append' option is not used (see paragraph Format of .props file). Check the 'Date Modified' to make sure it's the right file. Give the copy a suitable name, like appending the current date.
6. Perform the repair, upgrade or restoration of the database. For example, to restore to a backed-up file delete the original database file (the one you made a copy of in step 1 above) and rename the backed-up file to the name of the original database file. It is recommended to make a copy of the text in the log window (select+copy) for future references.
7. Start the SystemWeaver server application again, and check in the log window or file that no problem is detected during the start-up.

* A special case that should be considered carefully is when a SystemWeaver server appears to have stopped, due to memory problems. In such cases the server may continue to execute, at a very slow rate. The normal method of detecting activity, to look at the *CPU* value will then not be effective. Instead you should look at the *CPU Time* value during a period of minutes.

4.1.1 Stopping a SystemWeaver Server Application

All SystemWeaver servers are designed to be robust and to cope also with unexpected termination. However, since the behavior may depend on the specific database manager in use, specific behavior of the operating system, and network connection:

A SystemWeaver server should always be stopped in a controlled fashion since the database may need to perform actions as part of the shutdown.

Server Applications with a GUI, like the swTestServer have STOP buttons for the purpose of stopping the server in a controlled way.

The swServer, which runs with a minimal GUI, is stopped in a controlled way with the Window Exit button.

Server applications running as services with the swServerMonitorService should be stopped by the Windows Services tool (see section SystemWeaver as a Service). When the server is stopped this way, any connected clients will automatically be disconnected. The swServerMonitorService sends a stop request to the server application, but should the application not stop within 10 seconds, the monitor will terminate the application anyway. Thus, Step 2 above should always be performed appropriately.

Server applications running as background Windows processes, without any service active, must be stopped by the Windows Task Manager. Note that in this case there is no support for a controlled shut down, so the Steps 1 and 2 above must be performed appropriately.

4.2 Performing upgrades

To upgrade a SystemWeaver server to a new release the following steps should be executed:

1. The server must be stopped according to the section about Stopping and starting a SystemWeaver server.
2. Create a security back-up copy of the database, following the instructions in the section about Backup management.
3. The server application(s) is then replaced by the new release.
4. Depending on the version being upgraded from/to, the database may need to be upgraded using the swDatabaseManager (see section swDatabaseManager). You will be notified as part of the upgrade delivery whether or not this is the case.
5. The server is started again following the instructions in Stopping and starting a SystemWeaver server.
6. Connect to the server using the delivered SystemWeaver client applications, to assure that the server has been installed correctly. The client application that is used for the test should be installed using the instruction and procedures applied for all user installations to assure that the configuration that is tested is also the one that will be used by all users.

4.3 Feature installation

Some views and meta model constructs in SystemWeaver are managed as SystemWeaver Features. When the feature is installed, the related meta model constructs will be created automatically by the client by which the feature is installed, and all users will have the possibility to use the corresponding tool views in the SystemWeaver clients.

SystemWeaver features are installed using the **Configure the Explorer** option in the *File* tab, which opens the *Configure the Explorer* dialog ([4] swAdmin Help Manual). The installation is then done in the **Features** tab.

The only way to uninstall a feature is to remove the specific meta model construct.

Changes like installations of new features or meta data changes should always be performed in the following sequence:

1. Try out the change in a suitable test environment. The test environment is typically a "small" desktop installation where the effects of the change can be easily observed. You should always have the option to scrap the test environment in case you end up in trouble. During these steps, you should assure that installation instructions are used for the change (often these instructions are developed during the stage). This step may be performed multiple times until the change can be performed without trouble. This stage is typically performed by Systemite personnel or an organization that has acquired the necessary skill.
2. Try out the change in a copy of the production database. The copy should be executed in an exact replication of the real production database: be a recent copy, same version of SystemWeaver and MS Windows etc.
   The change procedure should follow the instruction used in the previous stage.
3. Perform the change in production database. This step should always be preceded by performing a backup of the production database.

4.4 Managing Licenses

Licenses come in the form of license keys stored in a license file.

License keys are loaded into the database using the swAdmin application. This procedure is described in [4] swAdmin Help Manual.

Licenses allow user accounts to be activated up to the available number of licenses.
Note that the licenses never affect the work that users can perform in SystemWeaver.

4.5 Troubleshooting

The paragraphs below describe some common problems and how they are handled. Some problems may relate to the operating system, the specific execution environment or the database software. Refer to documentation of these to solve related problems.

4.5.1 Compactation of large SQLite files

The swDatabaseManager includes an option called Vacuum SQLite database, which will minimize the SQLite database file. The benefit of this is that the sheer size of backup copies will become reduced and it will also improve performance to some degree.

4.5.2 Could not start sw server monitor service on local computer

This failure occurs in the case that the process does not have the right to start a service on the computer. Check that the proper rights are assigned to the service process.

4.5.3 Error 1067 when starting a service

The error message "Error 1067 The process terminated unexpectedly" can appear in the case that the services do not have write access to the directories where log files are created. SystemWeaver services are normally set to log on as

"Local System". Check file locations and access rights to assure the Local System has all the necessary rights.

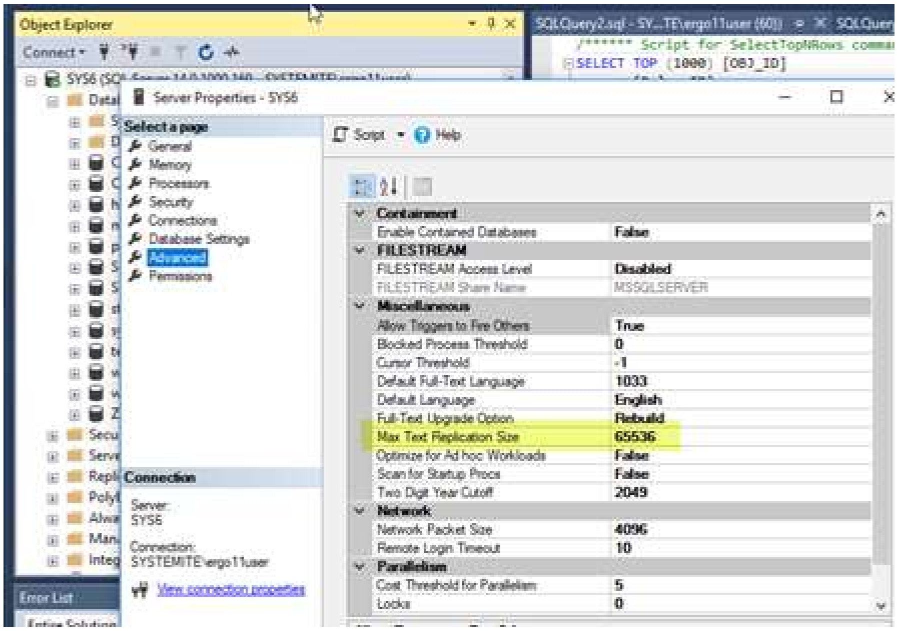
4.5.4 Error: Length of data to be replicated exceeds configured max

This error occurs when a user tries to add data to a replicated column or captured column in an MS SQL database that exceeds the allowed size.

Example error message

```
[FireDAC][Phys][ODBC][Microsoft][ODBC SQL Server Driver][SQL Server]Length of LOB data (70199) to
be replicated exceeds configured maximum 65536.
```

The problem resides on the MS SQL database server end. Consider adjusting the Max Text Replication Size setting.



```
/****** Script for SelectTopNRows comma
SELECT TOP (1000) [OBJ_ID]
```

Server:
SYS6

Connection:
SYSTEMITE\ergo11user

Containment	
Enable Contained Databases	False
FILESTREAM	
FILESTREAM Access Level	Disabled
FILESTREAM Share Name	MSSQLSERVER
Miscellaneous	
Allow Triggers to Fire Others	True
Blocked Process Threshold	0
Cursor Threshold	-1
Default Full-Text Language	1033
Default Language	English
Full-Text Upgrade Option	Rebuild
Max Text Replication Size	65536
Optimize for Ad hoc Workloads	False
Scan for Startup Procs	False
Two Digit Year Cutoff	2049
Network	
Network Packet Size	4096
Remote Login Timeout	10
Parallelism	
Cost Threshold for Parallelism	5
Locks	0

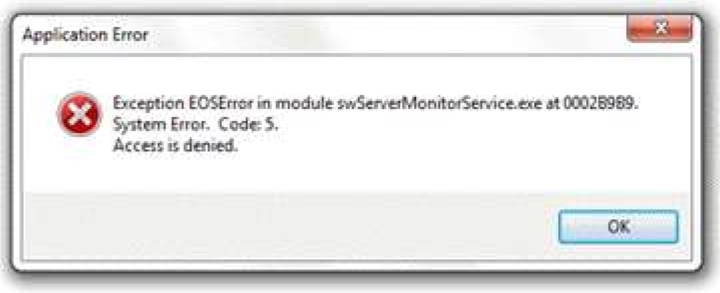
https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/configure-the-max-text-repl-size-server-configuration-option?view=sql-server-2017

4.5.5 Socket error 10053 from swNotificationServer

In the case of encountering socket error 10053 or 10013 from the swNotificationServer or the swTestNotificationServer, try to temporarily disable any anti-virus, or fire-wall or other security enhancing software since they can stop any outgoing email. Then configure the anti-virus software not to block SMTP requests (normally 587 although traditionally port 25 was used for this purpose) from processes named "swNotificationServer" and "swTestNotificationServer". Refer to your security software for proper procedures.

4.5.6 System Error. Code:5

This problem can occur when you try to install a service, and Administrator rights are required for installing services on the computer:


Application Error

```
Exception EOSError in module swServerMonitorService.exe at 0002B989.
System Error.  Code: 5.
Access is denied.
```

To avoid the problem, use the **Run as administrator** option in Windows when you do the installation.

4.5.7 SystemWeaver server dies immediately when started

In case the certificate files or the entire 'cert' directory is missing or with the wrong name the server will terminate immediately without any trace in the log files.

4.5.8 Windows Event Viewer

The Windows Event Viewer can be useful for debugging various faults related to installation, like the Error 1067 when starting a service.

The case illustrated below appeared when the installation folder had been removed so that a service could not start.

5 Backup management

The method to perform backup depends on the kind of SystemWeaver server/service installation; specifically, the method differs if SQLite, Oracle or SQL Server is being used. Always refer to the documentation from the supplier of the components you are using in your installation.

Note that the Backup utility in MS Windows will not perform backup of locked files, including database files, meaning that the database of a running SystemWeaver server, using a file based database, will actually never be backed up. The solution to this is to first stop the server, following the procedure according to Stopping and starting a SystemWeaver server. An alternative to stop the server is first to make a copy of the file, and then perform the backup on that file. Since there is always a risk of a concurrent database write transaction, the copied file may become corrupted. To minimize this risk, always perform the backup when there are few users (which may be difficult in a 24/7 situation) typically at night.
Backup routines should never be trusted without proper verification. The Administrator should therefore regularly assure that the backups can be restored, and that they really represent the intended database copy.

5.1 Runtime Backup using SQLite

The Runtime backup option for SQLite guarantees successful backup copies when users are logged in to the server and perform write operations. While the backup is performed, user will be able to browse SystemWeaver data, but write operations will lock the client until the backup operation has been completed. It is possible to use this functionality for all database installations, regardless of database type.

The EXE file "swServerCommand.exe" (refer to the section swServerCommand) uses a backup API call in the SystemWeaver server.

The "backup.bat" file supplied in SystemWeaver releases contains an example of how the backup functionality can be used:

```
swServerCommand backup ServerName=localhost ServerPort=1768
@echo off
rem swServerCommand backup ServerName=localhost ServerPort=1768 FileName="D:\backup.sqlite"
FileRepositoryFileName="D:\backup_files.sqlite"
if %ERRORLEVEL% NEQ 0 (
  echo Backup command returned error
  pause
  exit
)

pause
```

The parameters to the swServerCommand are described below.

The **backup** parameter specifies the backup operation.

The **ServerName** and **ServerPort** parameters identifies the server.

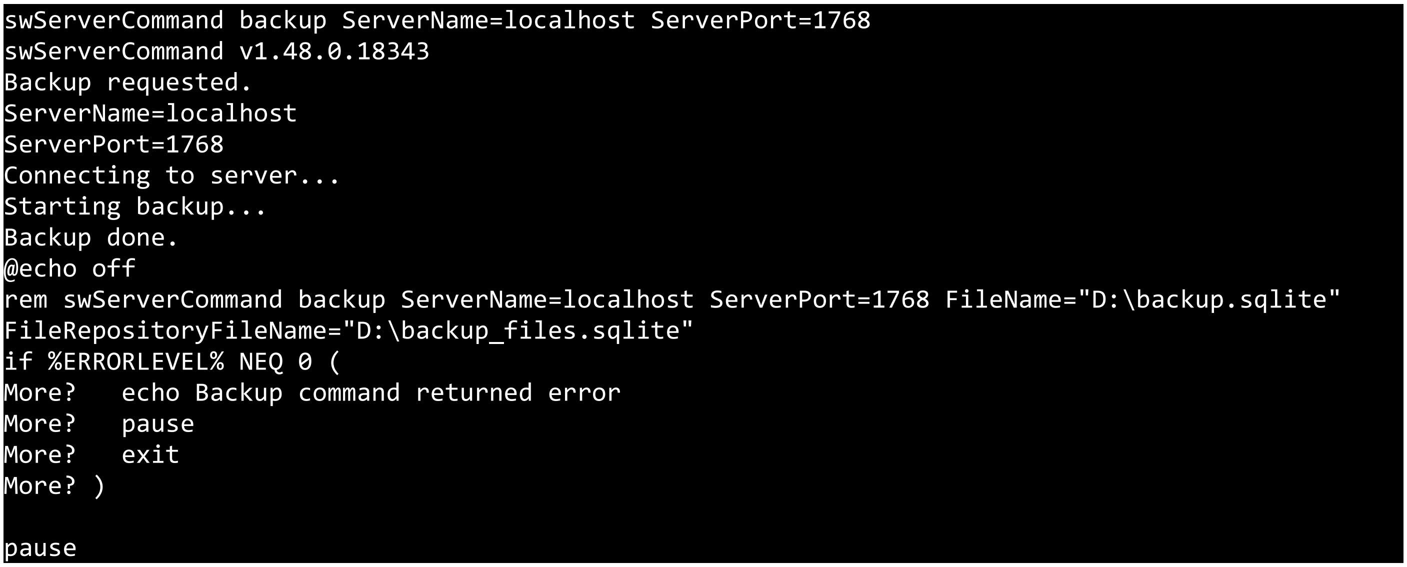
The **FileName** specifies the name of the backup file.
If no name is specified an automatic name will be used based on the name of the database file, current date, a random

number, and appended with "_backup", like: "MyDatabase_20160921_172932_959_backup.sqlite". The file will be created in the folder where the SQLite file is located.

The **FileRepositoryFileName** specifies the name of the backup file for the file repository database.
If no name is specified an automatic name is used, similar to the case of the database, but appended with "_backup_files.sqlite".
If a file based file repository is used, an index file backup will be created, appended with "_backup_files_index.db".

Note that if the FileName is specified, but not the FileRepositoryFileName, then the backup will only perform a backup of the database.

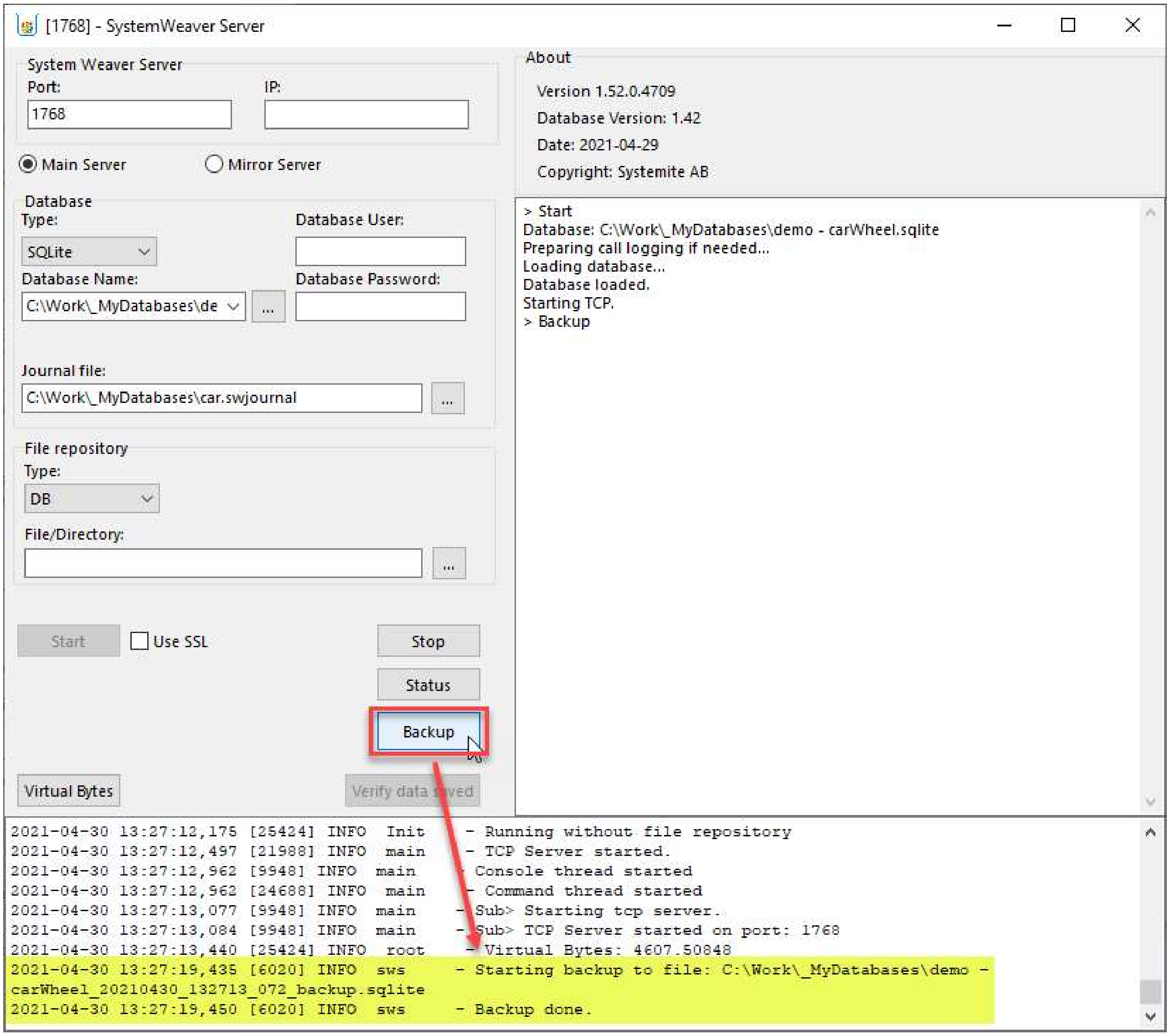
If a script according to the example above runs successfully the result in the command window should become:



```
swServerCommand backup ServerName=localhost ServerPort=1768
swServerCommand v1.48.0.18343
Backup requested.
ServerName=localhost
ServerPort=1768
Connecting to server...
Starting backup...
Backup done.
@echo off
rem swServerCommand backup ServerName=localhost ServerPort=1768 FileName="D:\backup.sqlite"
FileRepositoryFileName="D:\backup_files.sqlite"
if %ERRORLEVEL% NEQ 0 (
More?   echo Backup command returned error
More?   pause
More?   exit
More? )

pause
```

The figure below shows the backup feature as available in the swTestServer:



System Weaver Server

Port: 1768

IP:

● Main Server ○ Mirror Server

Database

Type: SQLite

Database User:

Database Name: C:\Work_MyDatabases\de

Database Password:

Journal file: C:\Work_MyDatabases\car.swjournal

File repository

Type: DB

File/Directory:

□ Use SSL

About

Version 1.52.0.4709

Database Version: 1.42

Date: 2021-04-29

Copyright: Systemite AB

```
> Start
Database: C:\Work\_MyDatabases\demo - carWheel.sqlite
Preparing call logging if needed...
Loading database...
Database loaded.
Starting TCP.
> Backup
```

```
2021-04-30 13:27:12,175 [25424] INFO  Init     - Running without file repository
2021-04-30 13:27:12,497 [21988] INFO  main     - TCP Server started.
2021-04-30 13:27:12,962 [9948] INFO  main     - Console thread started
2021-04-30 13:27:12,962 [24688] INFO  main     - Command thread started
2021-04-30 13:27:13,077 [9948] INFO  main     - Sub> Starting tcp server.
2021-04-30 13:27:13,084 [9948] INFO  main     - Sub> TCP Server started on port: 1768
2021-04-30 13:27:13,440 [25424] INFO  root     - Virtual Bytes: 4607.50848
2021-04-30 13:27:19,435 [6020] INFO  sws     - Starting backup to file: C:\Work\_MyDatabases\demo -
carWheel_20210430_132713_072_backup.sqlite
2021-04-30 13:27:19,450 [6020] INFO  sws     - Backup done.
```

In case an SQLite file repository is used the file backup file will contain the entire file repository.

If the file-based file repository is used, the backup file will contain only the file index. To get a backup index file in synch with the file repository, it is recommended that the backup of the file repository is performed at the same* time.

* Ideally at the same time, although a file repository backup from a later point of time will pose no problem, since additional files in the repository will not be detected by the file index.

5.2 Restoring a database

Follow the instructions according to Stopping and starting a SystemWeaver server to stop, and restart the server with the backup file.

6 User Administration

Refer to [2] SystemWeaver Access Rights and Administration for descriptions on how to use the swAdmin client to manage user accounts, and how to use the access control mechanisms of SystemWeaver.

The user organization is normally obliged by a license agreement or contract to the SystemWeaver distributor at all times to ensure that the number of users do not exceed the number of licenses. A user that has not been
The obligation is also controlled by the license management system built into the SystemWeaver server. The way to manage licenses is described in the section about Managing Licenses.
Refer to the contract for details applicable to your user organization.

7 Security Management

Security Management options are described in ref [2] SystemWeaver Access Rights and Administration, including recommendations regarding security.

The special "root" user password is set by default for a release or installation. In the event that the user organization does not want Systemite AB to maintain this password, it should be changed using the Admin account. The password for the "root" user account should then be stored and maintained in a secure place. The "root" account should never be used for normal operations, only for emergency operations, e.g., if the password to all admin accounts are lost.

For further details regarding the root account refer to [2] SystemWeaver Access Rights and Administration.

8 Managing distributions

The table below lists all computer files that may be part of client or server configurations.
Each column groups a number of applications typically used together for a specific purpose by a specific user, and the files that the applications depend on.

- The "Client" group contains what a common user will need.
- The "Architect" group contains what a SystemWeaver architect will typically need.
- The "Admin" group contains what a SystemWeaver administrator vill typically need.
- The "Server" group contains what is needed for a SystemWeaver main or mirror server installation.
- The "Test Server" group contains what is needed for various SystemWeaver server installations for test purposes.
- The "Notification Server" group contains what is needed for the SystemWeaver notification server installation.
- The "API" group contains what is needed for an application connecting to a SystemWeaver server using the SystemWeaver client API.
- The "Extensions API" group contains files for development of extension views in SystemWeaver.
- The "REST API" group contains what is needed for the SystemWeaver REST API installation.

All applications in a group should be located in the same folder as the utility and data files, and optionally all applications in the group could also be located in the same folder.

Blank cells mean that the inclusion is not required for the specific installation. "o" means that the inclusion may be used for the specific configuration, but not required.

	Group	Client	Architect	Admin	Server	Test Server	Notification Server	API	Extensions API	REST API	Configuration files	Description
File	Type*											
american.adm	dictionary file	x										See note below
ApplicationInsights	config										x	
backup	BAT				x							Refer to the section Runtime Backup using SQLite
CARoot**	PFX				x							Server certificate file. Only used with encryption variant.
DotNetZip	DLL				x							
Ionic.Zlib	DLL				x							
K4os.Compression.LZ4	DLL				x							
libeay32**	DLL	x	x		x	x	x					OpenSSL driver. In 32 or 64 bit version (despite "32" in name). Only used with encryption variant.
liblz4	DLL	x	x		x	x	x					
Microsoft.AI.Agent.Intercept	DLL				x							
Microsoft.AI.DependencyCollector	DLL				x							
Microsoft.AI.PerfCounterCollector	DLL				x							
Microsoft.AI.ServerTelemetryChannel	DLL				x							
Microsoft.AI.WindowsServer	DLL				x							

Microsoft.ApplicationInsights	DLL				x							
NLog	DLL				x							
NOTICES	TXT	x	x									
RemObjects.Hydra	DLL	x	x									
RemObjects.Hydra.WPF	DLL	x	x									
RVFUtility	DLL	x						x				Conversion between RTF and RVF text formats
RVFUtility64	DLL				x			x				64 bit version of *RVFUtility*
ssleay32**	DLL	x	x		x	x	x					OpenSSL driver. In 32 or 64 bit version (despite "32" in name). Only used with encryption variant.
swAdmin2	application			x								Refer to the section about the swAdmin
swArchitect	application		x									Refer to the section about the swArchitect
swConnectionTest	application				x	x						
swDatabaseManager	application				x							Refer to the section about the swDatabaseManager
swDBserver	application				x							
swDBServerScale	application											
swExplorer	HTML	x										Refer to the section about the swExplorer.html File
swExplorer	application	x										Refer to the section about the swExplorer
swExplorerContextHelp	XML	x	x	x								Index file for the swHelp
swExplorerExtensions	folder	x	x									Refer to the section about the swExplorer
swExplorerLauncher	utility	x										Refer to the section about the swExplorerLauncher
swExplorerLauncher	conf	x										Refer to the section about the swExplorerLauncher
swExplorerLauncherRegister	BAT	x										Refer to the section about the swExplorerLauncher
swExplorerLauncherUnregister	BAT	x										Refer to the section about the swExplorerLauncher
SWExtension.AttachmentSearch	DLL	x										
SWExtension.CompletePartTypes	DLL	x										
SWExtension.ComponentGraph	DLL	x										
SWExtension.MetaModelGraphs	DLL	x										
SWExtension.XhtmlDescription	DLL	x										
swHelp	XML	x	x	x								Help file
swHelpReader	utility	x	x	x								Help file viewer
swNotificationServer	PROPS						x					Refer to the section Format of .props file
swNotificationServer	email server						x					Refer to the section about the SystemWeaver Notification Server
swNotificationServer	INI						x					Refer to the section Format of swNotificationServer.ini file
swServer	INI					x						Refer to the section about the Format of swServer.ini file
swServer	PROPS					x						Refer to the section about the Format of .props file
swServerCommand	utility				x							
swServerMonitorService	PROPS				x							Refer to the section about the Format of .props file

swServerMonitorService	SETTINGS				x							Refer to the section about the Format of swServerMonitorService.settings file
swServerMonitorService	utility				x							
swServerMonitorService_install	BAT				x							
swServerMonitorService_uninstall	BAT				x							
swTestNotificationServer	email server					x	x					Refer to the section about the SystemWeaver Notification Server
swTestServer	server				x	x						
System.Diagnostics.DiagnosticSource	DLL				x							
Systemite.Lz4	DLL				x							
Systemite.SystemWeaver.Explorer	DLL	x	x									
Systemite.SystemWeaver.Explorer.Hydra	DLL	x	x									
Systemite.SystemWeaver.ExtensionControls	DLL	x	x									
Systemite.SystemWeaver.Graphs	DLL	x	x									
Systemite.SystemWeaver.Graphs.Common	DLL	x	x									
Systemite.SystemWeaver.TcpSubServer	NLOG				x							
Systemite.SystemWeaver.TcpSubServer	PDB				x							
Systemite.SystemWeaver.TcpSubServer	application				x							
SystemWeaver.ClientAPI	DLL				x			x		x		Compiled as Any CPU. Will work with either 32-bit or 64-bit.
SystemWeaver.ClientAPI	XML							x				
SystemWeaver.Common	DLL				x							
SystemWeaver.RestApi	EXE									x		
SystemWeaver.RestApi	XML									x		
SystemWeaver.RestAPI.exe	CONFIG									x		
SystemWeaverDatabaseManager	DLL											Embedded version of the swDatabaseManager
SystemWeaverEmbeddedServer	DLL							x				
SystemWeaverEmbeddedServer64	DLL							x				Embedded version of the server
SystemWeaverExtensionsAPI	XML								x			
SystemWeaverExtensionsAPI	DLL	x	x						x			
vcruntime140	DLL	x	x		x	x	x					
WebView2Loader	DLL	x	x									

Note that the following files must be located on the C drive of each client PC, due to technical reasons in Windows:

- The 'american.adm' spelling dictionary file

The 'swExplorerLauncher.exe', which supports the use of SystemWeaver item IDs as a URL in Windows, must be located in the same folder as the 'swExplorer.exe'.

In addition to these, there may be additional files for specific custom installations.

* The different file types are:

- application - a SystemWeaver client application. Windows file extension is .exe
- server - a SystemWeaver server application. Windows file extension is .exe
- utility - a small utility program. Windows file extension is .exe
- DLL - Windows DLL application extension. Windows file extension is .dll
- BAT - Windows BAT file. Windows file extension is .bat
- XML - configuration file in XML format
- INI - configuration/initialization file in text format
- PROPS - configuration file in log4d/log4net format
- PLUGINS - configuration file for the management of swExplorer extensions.
- PEM - Privacy Enhanced Mail is a Base64 encoded DER certificate.
- PDB - Program database (PDB) is a proprietary file format (developed by Microsoft) for storing debugging information about a program.

** The availability of the encryption option is restricted due to EU regulations. Please contact us for more information.

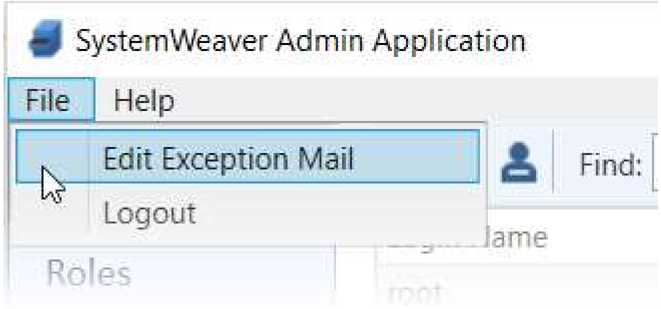
9 Managing Feedback and Error Reports

Systemite wants feedback on our applications. Feedback can be feature requests or defect reports.

Such feedback can be collected by an administrator and sent to support@systemite.se.

The swExplorer client also sends background, automatic error reports to us. Such automatic reports never reveal SystemWeaver or personal information and the sending is not noticeable by the user. The report only contains information about the involved software modules of SystemWeaver and the current Windows environment, including the computer name.
An example of an error report is provided in the sub section Sample error report.

Note that the automatic reports must be actively enabled using the swAdmin client, something Systemite kindly recommends. (see figure below)



s

9.1 Sample error report

```
------------------------------------------------------------------------------
Exception log with detailed tech info. Generated on 2015-10-05 15:32:14.
You may send it to the application vendor, helping him to understand what had happened.
 Application title: swexplorer
 Application file: \\sys4\Builds\2015-09-18 12.30 - 15198 - 2015R18 ReleaseKandidat\swExplorer.exe
 Application version:
------------------------------------------------------------------------------
Exception class: EJclFileVersionInfoError
Exception message: The file \\sys4\Builds\2015-09-18 12.30 - 15198 - 2015R18 ReleaseKandidat\swExplorer.exe does not
exist.
Exception address: 007D499B
------------------------------------------------------------------------------
Stack list, generated 2015-10-05 15:32:14
(003D3996){swExplorer.exe} [007D4996] JclFileUtils.TJclFileVersionInfo.Create (Line 5022, "JclFileUtils.pas" + 2) +
$1F
(01200ABA){swExplorer.exe} [01601ABA] swTab_Welcome.TFormTabWelcome.ShowAbout (Line 779, "swTab_Welcome.pas" + 3) +
$19
(011FF0BC){swExplorer.exe} [016000BC] swTab_Welcome.TFormTabWelcome.ActionAboutExecute (Line 234, "swTab_Welcome.pas"
+ 0) + $0
(000D54A3){swExplorer.exe} [004D64A3] System.Classes.TBasicAction.Execute + $F
(0013714A){swExplorer.exe} [0053814A] Vcl.ActnList.TCustomAction.Execute + $86
(000D52FB){swExplorer.exe} [004D62FB] System.Classes.TBasicActionLink.Execute + $13
(0091F7FE){swExplorer.exe} [00D207FE] dxBar.TdxBarItem.DoClick (Line 32033, "dxBar.pas" + 5) + $7
(0091F77B){swExplorer.exe} [00D2077B] dxBar.TdxBarItem.DirectClick (Line 32020, "dxBar.pas" + 5) + $7
(0092FA57){swExplorer.exe} [00D30A57] dxBar.TdxBarItemControl.ControlUnclick (Line 40128, "dxBar.pas" + 29) + $7
(0093336B){swExplorer.exe} [00D3436B] dxBar.TdxBarButtonControl.ControlUnclick (Line 42224, "dxBar.pas" + 11) + $4
(0093B768){swExplorer.exe} [00D3C768] dxBar.TCustomdxBarControl.DoLButtonUp (Line 46304, "dxBar.pas" + 18) + $6
(00946B1D){swExplorer.exe} [00D47B1D] dxBar.TdxBarControl.DoLButtonUp (Line 52057, "dxBar.pas" + 24) + $4
(0093A08C){swExplorer.exe} [00D3B08C] dxBar.TCustomdxBarControl.WMLButtonUp (Line 45601, "dxBar.pas" + 6) + $6
(0093E7CD){swExplorer.exe} [00D3F7CD] dxBar.TdxBarDockedControl.WMLButtonUp (Line 48092, "dxBar.pas" + 8) + $2
```

```
(00154681){swExplorer.exe} [00555681] Vcl.Controls.TControl.WndProc + $2BD
(00009136){swExplorer.exe} [0040A136] System.@IsClass + $E
(00009161){swExplorer.exe} [0040A161] System.@AsClass + $15
(008EF942){swExplorer.exe} [00CF0942] dxBar.ActiveBarManager (Line 9081, "dxBar.pas" + 1) + $5
(008FA15E){swExplorer.exe} [00CFB15E] dxBar.HideAllIfNeeded (Line 13801, "dxBar.pas" + 17) + $6
(00159229){swExplorer.exe} [0055A229] Vcl.Controls.TWinControl.WndProc + $5E9
(00009D40){swExplorer.exe} [0040AD40] System.TMonitor.TryEnter + $28
(000098C8){swExplorer.exe} [0040A8C8] System.TMonitor.Enter + $10
(0093AC2E){swExplorer.exe} [00D3BC2E] dxBar.TCustomdxBarControl.WndProc (Line 45961, "dxBar.pas" + 48) + $5
(00009768){swExplorer.exe} [0040A768] System.TMonitor.CheckOwningThread + $4
(00009A56){swExplorer.exe} [0040AA56] System.TMonitor.Exit + $6
(00009AA7){swExplorer.exe} [0040AAA7] System.TMonitor.Exit + $F
(0015885D){swExplorer.exe} [0055985D] Vcl.Controls.TWinControl.MainWndProc + $41
(00158A89){swExplorer.exe} [00559A89] Vcl.Controls.TWinControl.IsControlMouseMsg + $65
(000D6200){swExplorer.exe} [004D7200] System.Classes.StdWndProc + $14
(00159229){swExplorer.exe} [0055A229] Vcl.Controls.TWinControl.WndProc + $5E9
(008F3092){swExplorer.exe} [00CF4092] dxBar.TdxBarPopupControl.IsInternal (Line 10447, "dxBar.pas" + 0) + $2
(0093C9D7){swExplorer.exe} [00D3D9D7] dxBar.TCustomdxBarControl.GetEquivalentParentBar (Line 46949, "dxBar.pas" + 1) +
$4
(0093B95E){swExplorer.exe} [00D3C95E] dxBar.TCustomdxBarControl.CanProcessMouseMessage (Line 46370, "dxBar.pas" + 6) +
$2
(0093ABE3){swExplorer.exe} [00D3BBE3] dxBar.TCustomdxBarControl.WndProc (Line 45953, "dxBar.pas" + 40) + $5
(0094867F){swExplorer.exe} [00D4967F] dxBar.TdxBarControl.WndProc (Line 52931, "dxBar.pas" + 52) + $4
(00158848){swExplorer.exe} [00559848] Vcl.Controls.TWinControl.MainWndProc + $2C
(000D6200){swExplorer.exe} [004D7200] System.Classes.StdWndProc + $14
(002742DB){swExplorer.exe} [006752DB] Vcl.Forms.TApplication.ProcessMessage + $F3
(0027431E){swExplorer.exe} [0067531E] Vcl.Forms.TApplication.HandleMessage + $A
(00274651){swExplorer.exe} [00675651] Vcl.Forms.TApplication.Run + $C9
(01E69DC0){swExplorer.exe} [0226ADC0] swExplorer.swExplorer (Line 481, "" + 24) + $2
------------------------------------------------------------------------------
System   : Windows 7 Professional, Version: 6.1, Build: 1DB1, "Service Pack 1"
Processor: Intel,        Intel(R) Core(TM) i7-2640M CPU @ 2.80GHz, 2790 MHz MMX SSE SSE2 SSE3 SSSE3 SSE41 SSE42 64
bits DEP
Memory: 8073; free 3107
Display  : 1920x1200 pixels, 32 bpp
------------------------------------------------------------------------------
List of loaded modules:
[00400000] \\sys4\Builds\2015-09-18 12.30 - 15198 - 2015R18 ReleaseKandidat\swExplorer.exe
           (no module version info)
           Unit versioning information:
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\windows $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\windows $URL$ $Revision$ $Date$
             JCL\source\windows $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\windows $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\windows $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\windows $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
             JCL\source\windows $URL$ $Revision$ $Date$
             JCL\source\windows $URL$ $Revision$ $Date$
             JCL\source\windows $URL$ $Revision$ $Date$
             JCL\source\windows $URL$ $Revision$ $Date$
             JCL\source\windows $URL$ $Revision$ $Date$
             JCL\source\windows $URL$ $Revision$ $Date$
             JCL\source\common $URL$ $Revision$ $Date$
[10000000] C:\Program Files\WIDCOMM\Bluetooth Software\SysWOW64\BtMmHook.dll
           6.3.0.8800 - 6.3.0.8800
           Multimedia Keys Hook DLL
[11000000] \\sys4\Builds\2015-09-18 12.30 - 15198 - 2015R18 ReleaseKandidat\libeay32.dll
           (no module version info)
[12000000] \\sys4\Builds\2015-09-18 12.30 - 15198 - 2015R18 ReleaseKandidat\ssleay32.dll
           (no module version info)
[58300000] C:\Windows\system32\wpdshext.dll
           6.1.7601.18738 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Portable Devices Shell Extension
[58570000] C:\Windows\SysWOW64\PhotoMetadataHandler.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Photo Metadata Handler
[5C4B0000] C:\Windows\System32\cscobj.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           In-proc COM object used by clients of CSC API
[5C4E0000] C:\Windows\system32\browcli.dll
           6.1.7601.17887 - 6.1.7601.17887 (win7sp1_gdr.120704-0720)
           Browser Service Client DLL
[5C4F0000] C:\Windows\system32\dfscli.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows NT Distributed File System Client DLL
[5C500000] C:\Windows\System32\wcnapi.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows Connect Now - API Helper DLL
[5C520000] C:\Windows\SysWOW64\FunDisc.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Function Discovery Dll
[5C620000] C:\Windows\System32\fdwcn.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows Connect Now - Config Function Discovery Provider DLL
[5C640000] C:\Windows\system32\PortableDeviceApi.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows Portable Device API Components
[5C6D0000] C:\Windows\system32\NetworkExplorer.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Network Explorer
[5C8E0000] C:\Windows\system32\FirewallAPI.dll
           6.1.7600.16385 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           Windows Firewall API
[5C980000] C:\Windows\system32\SHFolder.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Shell Folder Service
[5CB40000] C:\Windows\SysWOW64\fdWNet.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Function Discovery WNet Provider Dll
[5CB50000] C:\Windows\SysWOW64\fdproxy.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Function Discovery Proxy Dll
[5CBA0000] C:\Windows\system32\SHDOCVW.dll
           6.1.7601.18222 - 6.1.7601.18222 (win7sp1_gdr.130725-1535)
           Shell Doc Object and Control Library
[5CBD0000] C:\Windows\system32\SearchFolder.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           SearchFolder
[5CE20000] C:\Windows\system32\NetworkItemFactory.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           NetworkItem Factory
[5CE30000] C:\Windows\system32\IconCodecService.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Converts a PNG part of the icon to a legacy bmp icon
[5CE40000] C:\Windows\SysWOW64\thumbcache.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Microsoft Thumbnail Cache
[5CE60000] C:\Windows\System32\ntlanman.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Microsoft® Lan Manager
[5CFB0000] C:\Windows\SysWOW64\Kerberos.DLL
           6.1.7601.18939 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Kerberos Security Package
[5D040000] C:\Windows\system32\logoncli.dll
           6.1.7601.17514 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           Net Logon Client DLL
[5D070000] C:\Windows\system32\adsldp.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           ADs LDAP Provider DLL
[5D0A0000] C:\Windows\system32\adsldpc.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           ADs LDAP Provider C DLL
[5D0E0000] C:\Program Files (x86)\Common Files\SYSTEM\MSMAPI\1033\MSMAPI32.DLL
           14.0.6122.5000 - 14.0.6122.5000
           Extended MAPI 1.0 for Windows NT
[5D180000] C:\Windows\system32\IdnDL.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Downlevel DLL
[5D6A0000] C:\Windows\System32\StructuredQuery.dll
           7.0.7601.17514 - 7.00.7601.17514 (win7sp1_rtm.101119-1850)
           Structured Query
[5DAE0000] C:\Program Files (x86)\Internet Explorer\ieproxy.dll
           11.0.9600.18015 - 11.00.9600.18015 (winblue_ltsb.150814-1859)
           IE ActiveX Interface Marshaling Library
[5DD50000] C:\Windows\SysWOW64\ATL.DLL
           3.5.2284.0 - 3.05.2284
           ATL Module for Windows XP (Unicode)
[5EA30000] C:\Windows\system32\mapi32.dll
           1.0.2536.0 - 1.0.2536.0 (win7_rtm.090713-1255)
           Extended MAPI 1.0 for Windows NT
[5EA80000] C:\Windows\system32\wkscli.dll
           6.1.7601.17514 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           Workstation Service Client DLL
[5EA90000] C:\Windows\system32\netutils.dll
           6.1.7601.17514 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           Net Win32 API Helpers DLL
[5EAE0000] C:\Windows\SysWOW64\actxprxy.dll
           6.1.7601.17514 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           ActiveX Interface Marshaling Library
[5EB30000] C:\Program Files (x86)\Common Files\microsoft shared\ink\tiptsf.dll
           6.1.7601.18512 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Tablet PC Input Panel Text Services Framework
[5EB90000] C:\Windows\system32\mssprxy.dll
           7.0.7600.16385 - 7.00.7600.16385 (win7_rtm.090713-1255)
           Microsoft Search Proxy
[61950000] C:\Program Files (x86)\Microsoft Office\Office14\olmapi32.dll
           14.0.7155.5000 - 14.0.7155.5000
           Extended MAPI 1.0 for Windows NT
[61C90000] C:\Program Files (x86)\Common Files\Microsoft Shared\OfficeSoftwareProtectionPlatform\OSPPC.DLL
           14.0.370.400 - 14.0.0370.400 (longhorn(wmbla).090811-1833)
           Office Software Licensing Client Dll
[61D80000] C:\Program Files (x86)\Common Files\Microsoft Shared\office14\1033\MSOINTL.DLL
           14.0.7139.5000 - 14.0.7139.5000
           Microsoft Office 2010 component
[61FF0000] C:\Windows\System32\DAVHLPR.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           DAV Helper DLL
[62000000] C:\Windows\System32\davclnt.dll
           6.1.7601.18912 - 6.1.7601.18912 (win7sp1_gdr.150701-0844)
           Web DAV Client DLL
[62020000] C:\Program Files (x86)\Common Files\Microsoft Shared\office14\MSORES.DLL
           14.0.7109.5000 - 14.0.7109.5000
           Microsoft Office 2010 component
[66FC0000] C:\Program Files (x86)\Common Files\Microsoft Shared\office14\mso.dll
           14.0.7153.5000 - 14.0.7153.5000
           Microsoft Office 2010 component
[68270000] C:\Windows\system32\GPAPI.dll
           6.1.7601.18711 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Group Policy Client API
[68290000] C:\Windows\SysWOW64\bcryptprimitives.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows Cryptographic Primitives Library
[682D0000] C:\Windows\SysWOW64\schannel.dll
           6.1.7601.18939 - 6.1.7601.18939 (win7sp1_gdr.150722-0600)
           TLS / SSL Security Provider
[68310000] C:\Windows\system32\DUI70.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows DirectUI Engine
[683D0000] C:\Windows\system32\DUser.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows DirectUser Engine
[68400000] C:\Windows\system32\explorerframe.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           ExplorerFrame
[68660000] C:\Windows\system32\D3D10Warp.dll
           6.2.9200.17033 - 6.2.9200.17033 (win8_gdr.140615-1509)
           Direct3D 10 Rasterizer
[68850000] C:\Windows\system32\d3d11.dll
           6.2.9200.16570 - 6.2.9200.16570 (win8_gdr.130327-1526)
           Direct3D 11 Runtime
[689D0000] C:\Windows\system32\DWrite.dll
           6.2.9200.17461 - 6.2.9200.16492 (win8_gdr_oobssr.130113-0015)
           Microsoft DirectX Typography Services
[68B10000] C:\Windows\SysWOW64\mshtml.dll
           11.0.9600.18015 - 11.00.9600.18015 (winblue_ltsb.150814-1859)
           Microsoft (R) HTML Viewer
[69EA0000] C:\Windows\system32\api-ms-win-downlevel-shlwapi-l2-1-0.dll
           6.2.9200.16492 - 6.2.9200.16492 (win8_gdr_oobssr.130113-0015)
           ApiSet Stub DLL
[6A4C0000] C:\Windows\system32\LINKINFO.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows Volume Tracking
[6A550000] C:\Windows\system32\credssp.dll
           6.1.7601.18939 - 6.1.7601.18939 (win7sp1_gdr.150722-0600)
           Credential Delegation Security Package
[6A560000] C:\Windows\system32\WINSTA.dll
           6.1.7601.18540 - 6.1.7601.18540 (win7sp1_gdr.140716-1508)
           Winstation Library
[6B8D0000] C:\Windows\SysWOW64\ieframe.dll
           11.0.9600.18015 - 11.00.9600.18015 (winblue_ltsb.150814-1859)
           Internet Browser
[6C520000] C:\Windows\system32\WindowsCodecs.dll
           6.2.9200.17251 - 6.2.9200.17251 (win8_gdr.150128-1514)
           Microsoft Windows Codecs Library
[6CAA0000] C:\Windows\system32\oleacc.dll
           7.0.0.0 - 7.0.0.0 (win7sp1_gdr.110826-1504)
           Active Accessibility Core Component
[6CE60000] C:\Windows\system32\NLAapi.dll
           6.1.7601.18685 - 6.1.7601.18685 (win7sp1_gdr.141205-1623)
           Network Location Awareness 2
[6CF80000] C:\Windows\System32\bcrypt.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows Cryptographic Primitives Library (Wow64)
[6D2A0000] C:\Windows\WinSxS\x86_microsoft.vc90.crt_1fc8b3b9a1e18e3b_9.0.30729.6161_none_50934f2ebcb7eb57\MSVCP90.dll
           9.0.30729.6161 - 9.00.30729.6161
           Microsoft® C++ Runtime Library
[6D400000] C:\Windows\system32\SXS.DLL
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Fusion 2.5
[6E2B0000] C:\Windows\system32\ntshrui.dll
           6.1.7601.17755 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Shell extensions for sharing
[6E320000] C:\Windows\system32\EhStorShell.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows Enhanced Storage Shell Extension DLL
[6E360000] C:\Windows\System32\msxml6.dll
           6.30.7601.18980 - 6.30.7601.18980
           MSXML 6.0 SP3
[6EA80000] C:\Program Files\TortoiseSVN\bin\libsvn_tsvn32.dll
           1.7.8.43435 - 1.7.8
           Subversion library dll built for TortoiseSVN
[6ED80000] C:\Program Files\TortoiseSVN\bin\TortoiseSVN32.dll
           1.7.11.23600 - 1, 7, 11, 23600
           TortoiseSVN shell extension client
[6EDF0000] C:\Windows\System32\npmproxy.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Network List Manager Proxy
[6EE00000] C:\Program Files (x86)\Common Files\TortoiseOverlays\TortoiseOverlays.dll
           1.1.3.21564 - 1, 1, 3, 21564
           TortoiseSVN overlay handler shim
[6F110000] C:\Windows\system32\MPR.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Multiple Provider Router DLL
[6F150000] C:\Program Files\TortoiseSVN\bin\TortoiseStub32.dll
           1.7.11.23600 - 1, 7, 11, 23600
           TortoiseSVN shell extension client
[6F420000] C:\Windows\system32\wsock32.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows Socket 32-Bit DLL
[6F5D0000] C:\Program Files (x86)\Common Files\Microsoft Shared\office14\riched20.dll
           14.0.7155.5000 - 14.0.7155.5000
           RichEdit Version 6.0
[6F770000] C:\Windows\system32\PROPSYS.dll
           7.0.7601.17514 - 7.00.7600.16385 (win7_rtm.090713-1255)
           Microsoft Property System
[6F870000] C:
\Windows\WinSxS\x86_microsoft.windows.gdiplus_6595b64144ccf1df_1.1.7601.18946_none_72d45ee78666ea32\gdiplus.dll
           6.1.7601.18946 - 6.1.7601.18946 (win7sp1_gdr.150730-0600)
           Microsoft GDI+
[6FA30000] C:\Windows\system32\winmm.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           MCI API DLL
[6FA80000] C:\Windows\system32\msimg32.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           GDIEXT Client DLL
[6FA90000] C:\Windows\system32\wtsapi32.dll
           6.1.7601.17514 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           Windows Remote Desktop Session Host Server SDK APIs
[6FAA0000] C:\Windows\system32\MLANG.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Multi Language Support DLL
[6FAF0000] C:\Windows\system32\slc.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Software Licensing Client Dll
[6FB00000] C:\Windows\system32\cscapi.dll
           6.1.7601.17514 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           Offline Files Win32 API
[6FB10000] C:\Windows\system32\srvcli.dll
           6.1.7601.17514 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           Server Service Client DLL
[6FC60000] C:\Windows\SysWOW64\api-ms-win-downlevel-shell32-l1-1-0.dll
           6.2.9200.16492 - 6.2.9200.16492 (win8_gdr_oobssr.130113-0015)
           ApiSet Stub DLL
[6FC70000] C:\Program Files\TortoiseSVN\bin\libsasl32.dll
           2.1.24.0 - 2.1.24
[6FC90000] C:\Program Files\TortoiseSVN\bin\intl3_tsvn32.dll
           0.14.6.0 - 0.14.6
           LGPLed libintl for Windows NT/2000/XP and Windows 95/98/ME
[6FCA0000] C:\Program Files\TortoiseSVN\bin\libaprutil_tsvn32.dll
           1.3.12.0 - 1.3.12
           Apache Portable Runtime Utility Library
[6FCE0000] C:\Program Files\TortoiseSVN\bin\libapr_tsvn32.dll
           1.4.6.0 - 1.4.6
           Apache Portable Runtime Library
[6FD20000] C:\Windows\system32\uxtheme.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Microsoft UxTheme Library
[6FDA0000] C:\Windows\WinSxS\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.7601.18837_none_41e855142bd5705d\comctl32.dll
           6.10.7601.18837 - 6.10 (win7_rtm.090713-1255)
           User Experience Controls Library
[70610000] C:\Windows\system32\EhStorAPI.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows Enhanced Storage API
[70CD0000] C:\Windows\system32\zipfldr.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Compressed (zipped) Folders
[70DD0000] C:\Windows\system32\PortableDeviceTypes.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows Portable Device (Parameter) Types Component
[70E20000] C:\Windows\system32\dtsh.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Detection and Sharing Status API
[70EE0000] C:\Windows\system32\windowscodecsext.dll
           6.2.9200.16492 - 6.2.9200.16492 (win8_gdr_oobssr.130113-0015)
           Microsoft Windows Codecs Extended Library
[70F30000] C:\Windows\system32\dwmapi.dll
           6.1.7601.18917 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Microsoft Desktop Window Manager API
[70FB0000] C:\Windows\system32\NTDSAPI.DLL
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Active Directory Domain Services API
[710A0000] C:\Windows\System32\drprov.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Microsoft Remote Desktop Session Host Server Network Provider
[710C0000] C:\Windows\system32\ACTIVEDS.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           ADs Router Layer DLL
[71140000] C:\Windows\system32\dxgi.dll
           6.2.9200.16492 - 6.2.9200.16492 (win8_gdr_oobssr.130113-0015)
           DirectX Graphics Infrastructure
[71190000] C:\Windows\system32\d2d1.dll
           6.2.9200.16765 - 6.2.9200.16492 (win8_gdr_oobssr.130113-0015)
           Microsoft D2D Library
[71940000] C:\Windows\system32\msimtf.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Active IMM Server DLL
[71C50000] C:\Windows\system32\xmllite.dll
           1.3.1001.0 - 1.3.1001.0
           Microsoft XmlLite Library
[72180000] C:\Windows\system32\wshbth.dll
           6.1.7601.17514 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           Windows Sockets Helper DLL
[72190000] C:\Windows\System32\winrnr.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           LDAP RnR Provider DLL
[721A0000] C:\Windows\system32\pnrpnsp.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           PNRP Name Space Provider
[721C0000] C:\Windows\system32\napinsp.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           E-mail Naming Shim Provider
[721D0000] C:\Windows\system32\msls31.dll
           3.10.349.0 - 3.10.349.0
           Microsoft Line Services library file
[72300000] C:\Windows\system32\ncrypt.dll
           6.1.7601.18939 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows cryptographic library
[72340000] C:\Windows\system32\cryptdll.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Cryptography Manager
[72370000] C:\Windows\system32\dsrole.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           DS Role Client DLL
[72400000] C:\Windows\system32\dhcpcsvc6.DLL
```

```
           6.1.7601.17970 - 6.1.7601.17970 (win7sp1_gdr.121009-0412)
           DHCPv6 Client
[72410000] C:\Windows\system32\Fwpuclnt.dll
           6.1.7601.18283 - 6.1.7600.16385 (win7_rtm.090713-1255)
           FWP/IPsec User-Mode API
[725D0000] C:\Windows\system32\rasadhlp.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Remote Access AutoDial Helper
[725E0000] C:\Program Files (x86)\Common Files\Microsoft Shared\Windows Live\WLIDNSP.DLL
           7.250.4225.0 - 7.250.4225.0
           Microsoft® Windows Live ID Namespace Provider
[72610000] C:\Windows\system32\DNSAPI.dll
           6.1.7601.17570 - 6.1.7600.16385 (win7_rtm.090713-1255)
           DNS Client API DLL
[72670000] C:\Windows\System32\wship6.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Winsock2 Helper DLL (TL/IPv6)
[72680000] C:\Windows\system32\olepro32.dll
           6.1.7601.17514 - 6.1.7601.17514
[726A0000] C:\Windows\system32\oledlg.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           OLE User Interface Support
[726C0000] C:\Windows\system32\winspool.drv
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows Spooler Driver
[72DC0000] C:\Windows\System32\wshtcpip.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Winsock2 Helper DLL (TL/IPv4)
[72DD0000] C:\Windows\system32\apphelp.dll
           6.1.7601.18777 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Application Compatibility Client Library
[72E20000] C:\Windows\system32\mswsock.dll
           6.1.7601.18254 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Microsoft Windows Sockets 2.0 Service Provider
[73050000] C:\Windows\system32\dhcpcsvc.DLL
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           DHCP Client Service
[73070000] C:\Windows\system32\api-ms-win-downlevel-advapi32-l2-1-0.dll
           6.2.9200.16492 - 6.2.9200.16492 (win8_gdr_oobssr.130113-0015)
           ApiSet Stub DLL
[73170000] C:\Windows\system32\ntmarta.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows NT MARTA provider
[732D0000] C:\Windows\system32\secur32.dll
           6.1.7601.18939 - 6.1.7601.18939 (win7sp1_gdr.150722-0600)
           Security Support Provider Interface
[732E0000] C:\Windows\system32\sfc_os.DLL
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows File Protection
[732F0000] C:\Windows\SysWOW64\SFC.DLL
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows File Protection
[73470000] C:\Windows\system32\webio.dll
           6.1.7601.17725 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           Web Transfer Protocols API
[734C0000] C:\Windows\system32\WINHTTP.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows HTTP Services
[73770000] C:\Windows\system32\RpcRtRemote.dll
           6.1.7601.17514 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           Remote RPC Extension
[73780000] C:\Windows\system32\rsaenh.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Microsoft Enhanced Cryptographic Provider
[737C0000] C:\Windows\system32\CRYPTSP.dll
           6.1.7601.18741 - 6.1.7601.18741 (win7sp1_gdr.150202-1526)
           Cryptographic Service Provider API
[73F60000] C:\Windows\system32\msi.dll
           5.0.7601.18896 - 5.0.7601.18896
           Windows Installer
[743B0000] C:\Windows\system32\version.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Version Checking and File Installation Libraries
[743C0000] C:\Windows\system32\WINNSI.DLL
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Network Store Information RPC interface
[743D0000] C:\Windows\system32\iphlpapi.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           IP Helper API
[743F0000] C:\Windows\system32\MSVCR100.dll
           10.0.40219.325 - 10.00.40219.325
           Microsoft® C Runtime Library
[744B0000] C:\Windows\system32\MSVCP100.dll
           10.0.40219.325 - 10.00.40219.325
           Microsoft® C Runtime Library
[746E0000] C:\Windows\WinSxS\x86_microsoft.vc90.crt_1fc8b3b9a1e18e3b_9.0.30729.6161_none_50934f2ebcb7eb57\MSVCR90.dll
           9.0.30729.6161 - 9.00.30729.6161
           Microsoft® C Runtime Library
[75110000] C:\Windows\syswow64\CRYPTBASE.dll
           6.1.7601.18939 - 6.1.7601.18939 (win7sp1_gdr.150722-0600)
           Base cryptographic API DLL
[75120000] C:\Windows\syswow64\SspiCli.dll
           6.1.7601.18939 - 6.1.7601.18939 (win7sp1_gdr.150722-0600)
           Security Support Provider Interface
[75180000] C:\Windows\syswow64\GDI32.dll
           6.1.7601.18898 - 6.1.7601.18898 (win7sp1_gdr.150617-0600)
           GDI Client DLL
[75270000] C:\Windows\syswow64\comdlg32.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Common Dialogs DLL
[752F0000] C:\Windows\syswow64\imagehlp.dll
           6.1.7601.18288 - 6.1.7601.18288 (win7sp1_gdr.131018-1533)
           Windows NT Image Helper
[75320000] C:\Windows\syswow64\USERENV.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Userenv
[75340000] C:\Windows\syswow64\DEVOBJ.dll
           6.1.7601.17621 - 6.1.7601.17621 (win7sp1_gdr.110523-2108)
           Device Information Set DLL
[75360000] C:\Windows\system32\IMM32.DLL
           6.1.7601.17514 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           Multi-User Windows IMM32 API Client DLL
[753C0000] C:\Windows\syswow64\api-ms-win-downlevel-ole32-l1-1-0.dll
           6.2.9200.16492 - 6.2.9200.16492 (win8_gdr_oobssr.130113-0015)
           ApiSet Stub DLL
[753D0000] C:\Windows\syswow64\urlmon.dll
           11.0.9600.18015 - 11.00.9600.18015 (winblue_ltsb.150814-1859)
           OLE32 Extensions for Win32
[75520000] C:\Windows\syswow64\CLBCatQ.DLL
           2001.12.8530.16385 - 2001.12.8530.16385 (win7_rtm.090713-1255)
           COM+ Configuration Catalog
[755B0000] C:\Windows\syswow64\SHLWAPI.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Shell Light-weight Utility Library
[75610000] C:\Windows\syswow64\msvcrt.dll
           7.0.7601.17744 - 7.0.7601.17744 (win7sp1_gdr.111215-1535)
           Windows NT CRT DLL
[756C0000] C:\Windows\syswow64\SETUPAPI.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows Setup API
[75860000] C:\Windows\syswow64\oleaut32.dll
           6.1.7601.18679 - 6.1.7601.18679
[758F0000] C:\Windows\syswow64\api-ms-win-downlevel-user32-l1-1-0.dll
           6.2.9200.16492 - 6.2.9200.16492 (win8_gdr_oobssr.130113-0015)
           ApiSet Stub DLL
[75900000] C:\Windows\syswow64\profapi.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           User Profile Basic API
```

```
[75910000] C:\Windows\syswow64\iertutil.dll
           11.0.9600.18015 - 11.00.9600.18015 (winblue_ltsb.150814-1859)
           Run time utility for Internet Explorer
[75B50000] C:\Windows\syswow64\WINTRUST.dll
           6.1.7601.18839 - 6.1.7601.18839 (win7sp1_gdr.150427-0707)
           Microsoft Trust Verification APIs
[75C10000] C:\Windows\SysWOW64\sechost.dll
           6.1.7601.18869 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Host for SCM/SDDL/LSA Lookup APIs
[75C30000] C:\Windows\syswow64\kernel32.dll
           6.1.7601.18939 - 6.1.7601.18015 (win7sp1_gdr.121129-1432)
           Windows NT BASE API Client DLL
[75D40000] C:\Windows\syswow64\MSCTF.dll
           6.1.7601.18731 - 6.1.7600.16385 (win7_rtm.090713-1255)
           MSCTF Server DLL
[75E10000] C:\Windows\syswow64\CFGMGR32.dll
           6.1.7601.17621 - 6.1.7601.17621 (win7sp1_gdr.110523-2108)
           Configuration Manager DLL
[75E40000] C:\Windows\syswow64\MSASN1.dll
           6.1.7601.17514 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           ASN.1 Runtime APIs
[75E50000] C:\Windows\syswow64\RPCRT4.dll
           6.1.7601.18939 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Remote Procedure Call Runtime
[75F40000] C:\Windows\syswow64\PSAPI.DLL
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Process Status Helper
[75F50000] C:\Windows\syswow64\Normaliz.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Unicode Normalization DLL
[75F60000] C:\Windows\syswow64\ADVAPI32.dll
           6.1.7601.18939 - 6.1.7601.18869 (win7sp1_gdr.150525-0603)
           Advanced Windows 32 Base API
[76010000] C:\Windows\syswow64\KERNELBASE.dll
           6.1.7601.18939 - 6.1.7601.18015 (win7sp1_gdr.121129-1432)
           Windows NT BASE API Client DLL
[76060000] C:\Windows\syswow64\api-ms-win-downlevel-version-l1-1-0.dll
           6.2.9200.16492 - 6.2.9200.16492 (win8_gdr_oobssr.130113-0015)
           ApiSet Stub DLL
[76070000] C:\Windows\syswow64\api-ms-win-downlevel-shlwapi-l1-1-0.dll
           6.2.9200.16492 - 6.2.9200.16492 (win8_gdr_oobssr.130113-0015)
           ApiSet Stub DLL
[76080000] C:\Windows\syswow64\LPK.dll
           6.1.7601.18985 - 6.1.7601.18985 (win7sp1_gdr.150901-1452)
           Language Pack
[76090000] C:\Windows\syswow64\SHELL32.dll
           6.1.7601.18918 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           Windows Shell Common Dll
[76CE0000] C:\Windows\syswow64\WININET.dll
           11.0.9600.18015 - 11.00.9600.16428 (winblue_gdr.131013-1700)
           Internet Extensions for Win32
[76ED0000] C:\Windows\syswow64\NSI.dll
           6.1.7600.16385 - 6.1.7600.16385 (win7_rtm.090713-1255)
           NSI User-mode interface DLL
[76EE0000] C:\Windows\syswow64\WLDAP32.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Win32 LDAP API DLL
[76F30000] C:\Windows\syswow64\CRYPT32.dll
           6.1.7601.18839 - 6.1.7601.18839 (win7sp1_gdr.150427-0707)
           Crypto API32
[77060000] C:\Windows\syswow64\api-ms-win-downlevel-normaliz-l1-1-0.dll
           6.2.9200.16492 - 6.2.9200.16492 (win8_gdr_oobssr.130113-0015)
           ApiSet Stub DLL
[77070000] C:\Windows\syswow64\USER32.dll
           6.1.7601.17514 - 6.1.7601.17514 (win7sp1_rtm.101119-1850)
           Multi-User Windows USER API Client DLL
[77170000] C:\Windows\syswow64\ole32.dll
           6.1.7601.18915 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Microsoft OLE for Windows
```

```
[772D0000] C:\Windows\syswow64\USP10.dll
           1.626.7601.18454 - 1.0626.7601.18454 (win7sp1_gdr.140424-1533)
           Uniscribe Unicode script processor
[77370000] C:\Windows\syswow64\WS2_32.dll
           6.1.7601.17514 - 6.1.7600.16385 (win7_rtm.090713-1255)
           Windows Socket 2.0 32-Bit DLL
[77780000] C:\Windows\syswow64\api-ms-win-downlevel-advapi32-l1-1-0.dll
           6.2.9200.16492 - 6.2.9200.16492 (win8_gdr_oobssr.130113-0015)
           ApiSet Stub DLL
[777B0000] C:\Windows\SysWOW64\ntdll.dll
           6.1.7601.18939 - 6.1.7600.16385 (win7_rtm.090713-1255)
           NT Layer DLL
------------------------------------------------------------------------------
Active Controls hierarchy:
TFormMain "FormMain"
------------------------------------------------------------------------------
```

10 Documenting an installation

Some important recommendations for documenting a SystemWeaver installation are described in the following sections.
There are several reasons for performing the documentation:

- Documenting the used procedures facilitates and secures the proper execution of repeated operations
- Documenting the procedures means that it becomes more easy to perform the procedures for new Administrators, even if the original Administrators are not available
- In case of problems, the documentation can be used for back-tracing the causes of the problems

10.1 Documenting Procedures

An Administrator should document all operations and procedures in such a way that the proper administration can be secured despite unplanned events and situations like loss of passwords, serverbreakdown, absence of the administrator etc. The Document should include the following:

- Reference to current and past SystemWeaver configurations including versions, installation dates and protocols, and where these are stored.
- Procedures used for database upgrades, installing and starting SystemWeaver servers
- Network address of server(s)
- Physical Location of server(s) if relevant
- Key passwords like the "root" password, or references to secure location of such passwords
- Procedures for Back-up management
- Procedures for announcing and distributing new client installations
- Procedures for the creation and management of user accounts
- Other recurring procedures, like announcing planned server shutdown

10.2 Documenting changes

An Administrator should document all changes made to a SystemWeaver installation in a protocol for future reference, including the date of the change and who performed the change. Such changes typically include the following events:

- The first installation of a database and SystemWeaver server
- Any new releases of Servers that are installed or clients distributed to the user organization. The protocol should include all operations performed, and all applications used and installed, including their versions. All installations including the files and folder structure should be maintained for future reference.
- Other changes in the installation like the moving of SystemWeaver files in the windows file structure
- All restarts performed of the server, whether planned or unplanned
- Any problems affecting the operation of the server, including problems related to SystemWeaver applications, the operating environment, or the operating procedures
- Any upgrades of the database (using the swDatabaseManager), meta model or feature set (using the swModelManager or swExplorer). The protocol should ideally include all the logs produced by SystemWeaver clients and servers during the installation procedure, e.g. by copying the log content of the log window in swDatabaseManager.

Since changes to the meta model can be performed by the swArchitect role, any user given this role should be informed by the Administrator.

10.3 swExplorer.html File

The swExplorer Client has a built-in web browser as the first view that is opened to a user. This Home page view is available even before logging in to a server, so it is available even if the server is down. The browser is set up to open a file with the predefined name "swExplorer.html" at the directory of the exe file. The file would typically redirect users to an external location containing news regarding the operational status of the servers, etc, or provide information and links.

Below is an example of a "swExplorer.html" file that redirects the user to a web page with the URL "http://company.net/swInfoPages/ Infopage_main.htm", a web page that would typically be used for communication with the users. The web page could, for example, be used for downloading manuals and new client distribution, for Q&A lists, contact information, etc:

```
<HTML>
<head>
<title>SystemWeaver Redirect Page</title>
<meta http-equiv="REFRESH"
content="0;url=http://company.net/swInfoPages/Infopage_main.htm">
</HEAD>
</HTML>
```

Links will open in the user's default browser.

11 References

11.1 [1] SystemWeaver Explorer Help Manual

11.2 [2] SystemWeaver Access Rights and Administration

11.3 [3] SystemWeaver Architect Help Manual

11.4 [4] swAdmin Help Manual