

# *SC*

---

## **Service Connector**

### **Configuration and Operation Guide**

#### **SC\_4\_Operation\_E (Version V1.0)**

---

**This document describes topics related to the SC configuration, operation and troubleshooting.**

This document is a spiritual ownership of STABILIT Informatik AG, and must not be used or copied without a written allowance of this company. Unauthorized usage is illegal in terms of Chapter 23 / 5 UWG / Swiss law.

The information in this document is subject to change without notice and should not be construed as a commitment by STABILIT Informatik AG.

Copyright © 2011 by STABILIT Informatik AG  
All rights reserved.

All other logos, product names are trademarks of their respective owners.

**Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at**

**<http://www.apache.org/licenses/LICENSE-2.0>**

**Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and limitations under the License.**

This Document has been created with Microsoft Word 2003 (11) with template file C:\STABILIT\STANDARD\TEMPLATES\S\_REP\_E.DOT and printed at 11 March 2011 15:33.



## Identification

Project:	SC
Title:	Service Connector
Subtitle:	Configuration and Operation Guide
Version:	V1.0
Reference:	SC_4_Operation_E
Classification:	Public
Keywords:	Configuration, Operation, Troubleshooting
Comment:	This document describes topics related to the SC configuration, operation and troubleshooting.
Author(s):	STABILIT Informatik AG Jan Trnka, Daniel Schmutz, Joel Traber
Approval (Reviewed by):	Signature ..... Jan Trnka  Signature ..... Francisco Gonzalez  Signature ..... Walter Mörgeli
Audience:	Project team, Eurex development team
Distribution:	Public
Filename	c:\stabilit\projects\eurex\sc\documents\sc_4_operation_e.doc

## Revision History

Date	Version	Author	Description
11.02.2011	V1.0	Jan Trnka	Initial version





# Table of Contents

1	PREFACE.....	4
1.1	Purpose & Scope of this Document.....	4
1.2	Definitions & Abbreviations.....	4
1.3	External References.....	5
1.4	Typographical Conventions.....	5
1.5	Outstanding Issues.....	5
2	CONCEPTS.....	6
2.1	Session Service.....	6
2.2	Publishing Service.....	6
2.3	File Service.....	7
2.4	HTTP Redirection Service.....	7
2.5	SC Cascading.....	7
2.6	Message Caching.....	8
2.7	Message Fan-Out.....	8
2.8	SC Console.....	8
2.9	Web GUI.....	8
2.10	Restrictions.....	8
2.10.1	Load balancing.....	8
2.10.2	Failover.....	8
2.10.3	Security.....	9
2.10.4	Virus and Intrusion Protection.....	9
3	NETWORK.....	10
3.1	Connection Topology.....	11
3.1.1	Direct connection.....	11
3.1.2	Single cascade.....	11
3.1.3	Double cascade.....	12
4	INSTALLATION, UPDATE, REMOVAL.....	13
4.1	Installation.....	13
4.2	Update.....	13
4.3	Removal.....	13
5	CONFIGURATION.....	14
6	OPERATIONS.....	15
6.1	Starting SC.....	15
6.2	Stopping SC.....	15
6.3	Other console operations.....	15
6.4	Monitoring SC with the Web GUI.....	16
6.5	Monitoring SC with the J-Console or Jvisualvm.....	18
7	TROUBLESHOOTING.....	19
7.1	General.....	19
7.2	The most often situations.....	19
8	GLOSSARY.....	20
	INDEX.....	22

Tables

Table 1 Abbreviations & Definitions.....4  
Table 2 External references.....5  
Table 3 Typographical conventions .....5

Figures

Figure 1 Example of a connection (simplified).....10  
Figure 2 Example of a cascaded connection (simplified) .....10





# 1

# Preface

## 1.1 Purpose & Scope of this Document

This document describes issues related to the configuration, operation and troubleshooting of the Service Connector. Good understanding of network communication and the Service Connector concepts is prerequisite to this reading.

The SC message protocol is described in [1].

This document is particularly important to the operation & support staff..

## 1.2 Definitions & Abbreviations

Item / Term	Definition / Description
DMZ	Demilitarized Zone (Network segment between two firewalls)
DNS	Domain Name Server
DOS	Denial Of Service
GUI	Graphical User Interface
HTML	Hypertext Mark-up Language
HTTP	Hypertext Transport Protocol
HTTPS	HTTP over SSL, encrypted and authenticated transport protocol
IE	Internet Explorer – Microsoft Web Browser
J2EE	Java 2 Platform Enterprise Edition
Java	Programming language and run-time environment from SUN
JDK	Java Development Kit
Linux	Unix-like operating system, open source
Log4j	Standard logging tool used in Java
OpenVMS	HP Operating system
OSI	Open System Interconnection reference model <a href="http://en.wikipedia.org/wiki/OSI_model">http://en.wikipedia.org/wiki/OSI_model</a>
RPC	Remote Procedure Call
SCMP	SC Message Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer – secure communication protocol with encryption and authentication
SC	Service Connector
TCP/IP	Transmission Control Protocol / Internet Protocol
URL	Uniform Resource Locator
Web Service	software designed to support interoperable machine-to-machine interaction over a network
Web-GUI	Graphical User Interface running in Web-Browser built with Web components. Also called Thin-Client.
Windows	Microsoft operating system family
XML	Extensible Mark-up Language
XSD	XML Schema Definition
XSLT	Extensible Style Sheet Language Transformation

**Table 1 Abbreviations & Definitions**

## 1.3 External References

References	Item / Reference to other Document
[1]	SC_1_SCMP-V1.1_E

**Table 2 External references**

## 1.4 Typographical Conventions

Convention	Meaning
<i>text in italics</i>	<i>USP feature or functionality or USP related comments</i>
text in Courier font	code example
[ phrase ]	In syntax diagrams, indicates that the enclosed values are optional
{ phrase1   phrase2 }	In syntax diagrams, indicates that multiple possibilities exist.
...	In syntax diagrams, indicates a repetition of the previous expression

**Table 3 Typographical conventions**

The terminology used in this document may be somewhat different to other sources. The chapter Glossary includes a list of often used terms with the explanation of their meaning in this document.

## 1.5 Outstanding Issues

Following issues are outstanding at the time of the document release:

- 1.

## 2

## Concepts

Service Connector is a message oriented middleware connecting applications providing services to other applications consuming the services. It fits into an overall service oriented architecture.

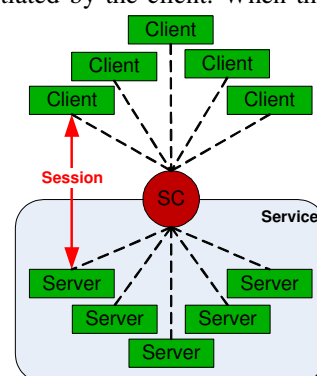
Service Connector supports following service types

### 2.1 Session Service

Session service allows statefull message exchange between requesting client application and server application implementing the service. All actions are initiated by the client. When the client starts the session, SC allocates a server instance for this session. The server must be ready at this time. Multiple server instances serving the same service can be pre-started. Serving multiple sessions or multiple services within one server instance is possible only for multisession servers.

Clients and servers are **not** connected directly to each other. The client and the server are only the logical communication end-points. The SC acts like a message broker and is never a direct executor of the service. Clients and servers may be distributed across any number of nodes in the network.

All services are independent on each other. Clients and servers may be distributed across any number of nodes in the network. The server application may request another service and so play the client role.

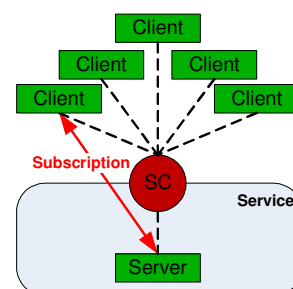


### 2.2 Publishing Service

Publishing service allows a server to send messages to many clients. In order to receive a published message, the client must subscribe to a service. The subscription mask allows fine granular definition of the client interest.

Clients and servers are **not** connected directly to each other. The application server which implements the service designates the contents of each message with a message mask and sends it to SC. When the message mask matches the client subscription mask, SC will deliver the message to the client. Multiple clients may subscribe for the same service at the same time and may get the same message. The publishing server does not know which client did will receive its messages and when. Sequence order of the published messages is preserved by SC.

Multiple server instances may publish their messages to the same service. (Not allowed when large messages are published!) All services are independent on each other. Clients and servers may be distributed across any number of nodes in the network. The server application may request another service and so play the client role.

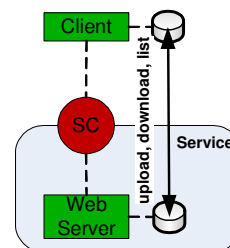


## 2.3 File Service

File service allows file exchange between client application and a web server. All actions are initiated by the client. The client can upload to a service or download file from a service or get a list of files in this service. Client and the Web Server are **not** connected directly to each other. The web server must define one flat directory for each service. Sub-directory structure is not supported. The same directory may be used by different services.

All services are independent on each other. Clients and web servers may be distributed across any number of nodes in the network.

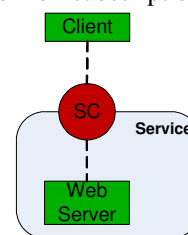
PHP is required on the Web Server in order to support file services. The provided PHP upload script may be adapted e.g. for sending notification mail after file upload.



## 2.4 HTTP Redirection Service

SC HTTP redirection service allows redirection of http traffic through the SC infrastructure to a designated target URL (web server). This is possible without a session or subscription context. It can be used by any application also by a browser. SC receives the http requests on a dedicated port and passes it to the configured server. It does not work like a HTTP proxy.

Client and the Web Server are **not** connected directly to each other. Clients and targets (web servers) may be distributed across any number of nodes in the network. The requested target URL must be resolved (by DNS or other means) by the underlying network infrastructure. SC does not provide any name resolution.



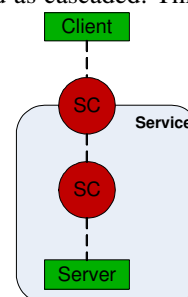
Multiple http redirection services may be defined to different targets, each one listening on a different TCP/IP port.

## 2.5 SC Cascading

Multiple SCs may be running in a network. Any SC service may be defined as cascaded. This means that the message traffic for such service is redirected to another SC. Cascading is fully transparent to the client. Clients can use services at any cascade level. Servers are allowed on the bottom level only.

Major benefits of cascading are:

- Improved security by using cascaded SC in a DMZ as a single access point for all local clients.
- Reduced communication costs by sharing infrastructure (VPN tunnel) by multiple clients
- Much lower outbound network traffic by utilizing message fan-out.
- Improved performance by utilizing SC cache near to the clients.
- Routing of services into other network segments.

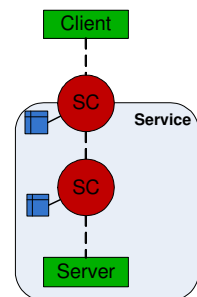


## 2.6 Message Caching

For session services SC implements caching of message sent from servers to clients. The cached message is available to all clients sending a request for it. Each service has its own cache and each cached message has an absolute expiration date and time. The cache is loaded by the first client requesting the messages. All subsequent clients will get the message from the cache, bypassing the server processing. Security is not affected because message exchange is possible only in a scope of a valid session.

The definition, which message should be cached, how it is identified and how long it is valid, is matter of agreement between the communicating parties.

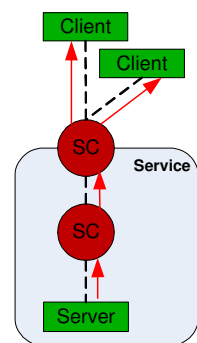
SC caching can be enabled or disabled per SC instance. In cascaded configurations caching is supported on cascaded session service. This allows caching near to the message consumer.



## 2.7 Message Fan-Out

For publishing services SC implements message fan-out in cascaded configurations. Message published by a server is distributed to the clients on the most upper cascade level. Between the SC nodes the message is transported only once. This allows efficient distribution of a message to large number of clients within a very short time.

The efficient usage of the fan-out is dependent on the design and structure of the subscription mask. It must contain items reflecting the cascade hierarchy.



## 2.8 SC Console

SC console is a tool which allows sending of simple management commands to SC from the command line. It may be used in scripts embedded into other management tools.

## 2.9 Web GUI

Web-GUI is a simple monitoring instrument showing the current SC activity. It can be disabled in the configuration. It utilizes http and can be used from any remote node having access to SC.

## 2.10 Restrictions

### 2.10.1 Load balancing

The SC does not provide any load balancing features. Established communication session will not be redirected to another server or service node during its life time.

### 2.10.2 Failover

The SC does not provide any failover features. Aborted communication must be re-established by the communicating partners. The client application must find out a service which is ready.

### **2.10.3 Security**

The SC does not implement any security feature. The environment where SC is used must provide all required authentication, authorization, encryption, tunneling etc. features. Message transport over https is not supported.

The IP address of the all nodes on the network path between the client and the server is available to the server when client starts a new session or the client subscribes. This can be used to authenticate and authorize the client when VPN tunnel is used.

### **2.10.4 Virus and Intrusion Protection**

The entire network where SC is used is assumed to be safe and secure. No virus protection is embedded in SC. Screening firewall can be used on any segment of the network. It is recommended to use SC within a DMZ.

SC is not designed to withstand network attacks like DOS or SYNC flood, or any other.

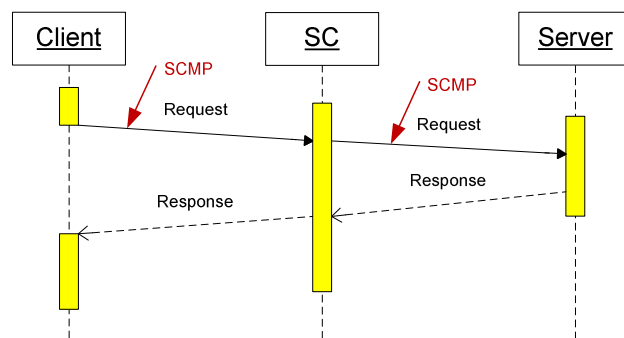
### 3

## Network

Clients and servers are not connected directly to each other. They are both connected to the SC. The Java API provided with SC and the communication between cascaded SCs uses a connection pools in order to save network resource.

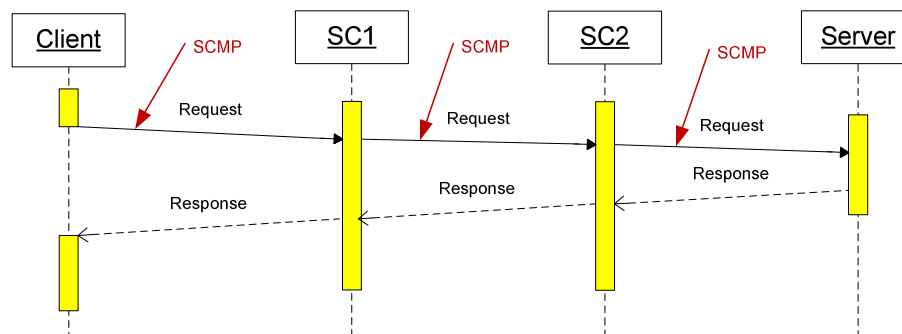
Connection pool is established for each two communication end points. There is no fix allocation of a connection to a certain session or subscription within the pool. Each request may use a different connection. Connections are created on demand and closed when they are idle for longer time. Connection pool can be configured to use TCP/IP or HTTP protocol. Using HTTP adds an extra overhead to SCMP but may be simpler to configure in foreign networks. SCMP over HTTP will pass a screening firewall placed between the end-points.

Sending of keep alive messages can be configured for each connection pool. This is required whenever a firewall is configured on this network segment.



**Figure 1 Example of a connection (simplified)**

Cascaded SCs are connected hierarchically.



**Figure 2 Example of a cascaded connection (simplified)**



## 3.1 Connection Topology

Client application, server application and the SC may reside on the same node or on separate nodes. The connection can either utilize HTTP protocol or direct TCP/IP communication. No assumption about the physical network topology is done. Multiple firewalls can be located on the path between the communicating partners. In order to demonstrate SC features, following connection topologies are shown:

1. Client  $\leftrightarrow$  SC  $\leftrightarrow$  Server = Direct connection
2. Client  $\leftrightarrow$  SC  $\leftrightarrow$  SC  $\leftrightarrow$  Server = Single cascade
3. Client  $\leftrightarrow$  SC  $\leftrightarrow$  SC  $\leftrightarrow$  SC  $\leftrightarrow$  Server = Double cascade

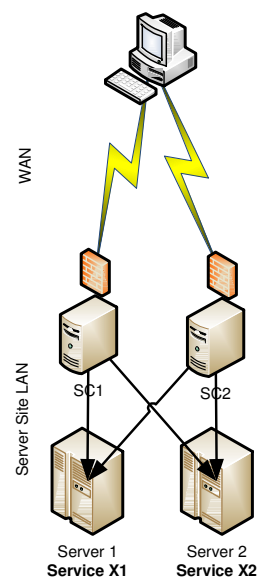
Other advanced cascaded topologies are possible.

### 3.1.1 Direct connection

This setup is suitable for applications with individual (mobile) clients connected directly to the service provider. In order to achieve higher application availability and scalability the server infrastructure can have redundant components.

The client can choose between connection to SC1 or 2 depending on their availability. Services X1 and X2 are identical and provided by servers 1 and 2. In case server 1 is not available service X2 can be used. Clients can use both X1 and X2 service and so distribute the load across all available servers. The logic to choose the access point and the appropriate service name is quite simple and must be implemented in the client.

The separation of SC nodes and the server nodes allows simple relocation of the services within the server site LAN and also allows scale-up the service performance by adding additional servers.



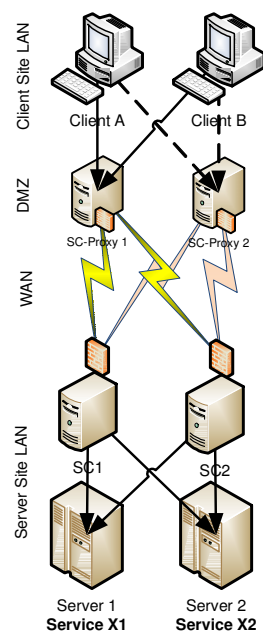
### 3.1.2 Single cascade

This setup is suitable for client applications running in enterprise environments. The clients are not connected directly to the service provider, but use an SC proxy placed within the enterprise DMZ as the local access point. In this way the enterprise security policies can be enforced. Utilization of the message cache in SC proxy can massively speed up the application. In order to achieve higher application availability SC proxy can be redundant. SC proxy can also be used to provide specific local services.

The client can choose between connection to SC proxy 1 or 2 depending on their availability. The logic to choose the access point and the appropriate service name is quite simple and must be implemented in the client.

Services X1 and X2 are identical and provided by server 1 and 2. In case server 1 is not available service X2 can be used. The network infrastructure must allow access between SC proxy 1, 2 and SC1 and 2.

The separation of SC nodes and the server nodes allows simple relocation of the services within the server site LAN and also allows scale-up the service performance by adding additional servers.



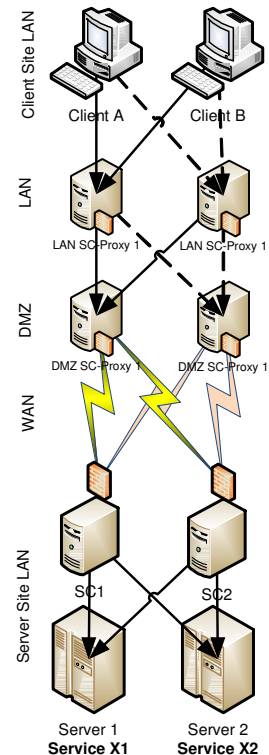
### 3.1.3 Double cascade

This setup is suitable for service providers wrapping the original service with some added value. The clients are not connected to the original service provider, but use an LAN SC proxy of the added value provider. Specific local services can be implemented here. The SC Proxy placed in the DMZ ensures the security policies and isolates the logical network segments. Utilization of message cache in the LAN SC proxy can massively speed up the application. In order to achieve higher application availability SC proxies can be redundant.

The client can choose between connection to LAN SC proxy 1 or 2 depending on their availability. The logic to choose the access point and the appropriate service name is quite simple and must be implemented in the client.

Services X1 and X2 are identical and provided by server 1 and 2. In case server 1 is not available service X2 can be used. The network infrastructure must allow access between SC proxy 1, 2 and SC1 and 2.

The separation of SC nodes and the server nodes allows simple relocation of the services within the server site LAN and also allows scale-up the service performance by adding additional servers.



## 4

# Installation, Update, Removal

SC is Open Source software under Apache License 2. It is available as a complete source code in the SVN repository <http://www.stabilit.ch/svn/repos/sc> and as a kit on <http://www.stabilit.ch/download/sc/>

### Kit Structure

The kit is available in two flavors:

- `sc-bin_V9.9-999.zip` kit containing only binaries
- `sc-src_V9.9-999.zip` kit containing binaries and additional resources required for development

The structure of the kit is described in the *readme.txt* file included in both zip-files.

### 4.1 Installation

For pre-requisites and products required by the specific SC version please see the *readme.txt* file included in the kit. These are usually:

- Java 1.6.18 or later
- Apache 2.0 or later (on node running file services)
- PHP 4.0 or later (on node running file services)

The installation is done by extracting the kit file into a directory of your choice. After the installation a verification procedure described in *readme.txt* file included in both zip-files.

### 4.2 Update

If you made changes to the SC configuration, then preserve these files before upgrade. The upgrade is simply done by extracting the kit file into a directory of the previous installation.

### Version compatibility

If you install new SC version you may see get error message of type **"Incompatible SC version nr."** This may happen when you clients or server has a higher version (is newer) then the SC (is older). The opposite situation (SC version is higher than the client/ server) is valid and the software will run without change.

For description of the compatibility issues please read SC\_0\_SCMP\_E.PDF Chapter: SCMP header Attributes - `scVersion`.

### 4.3 Removal

You must stop SC before you want to remove the software.

If you made changes to the SC configuration you must find out where did you put the:

- `log` directory
- `root.pidPath`
- `root.dumpPath`
- `cache.diskPath`

and delete all these directories.

Then you can delete the directory where did you extract the SC kit file.

## 5

## Configuration

The SC is configured by property files in the conf/ directory. These are:

- |                                   |                                       |
|-----------------------------------|---------------------------------------|
| – log4j-console.properties        | logging properties for sc-console     |
| – log4j-demo-client.properties    | logging properties for demo-client    |
| – log4j-demo-server.properties    | logging properties for demo-server    |
| – <b>log4j-sc.properties</b>      | <b>logging properties for sc</b>      |
| – <b>sc.properties.properties</b> | <b>sc configuration</b>               |
| – <b>sc-specific.properties</b>   | <b>site specific sc configuration</b> |

Tbd.

## 6

## Operations

This chapter describes the normal SC operations done by the operators.

Don't make changes to the scripts coming from the SC kit. Otherwise you will lose them after the next software update. It is recommended to wrap the scripts or to write your own scripts.

### 6.1 Starting SC

The Service connector is started with the script `bin/start-sc.bat` on Windows or `start-sc.sh` on unix / linux platforms.

It is possible to declare environment variables which are passed to SC where they can be used for configuration. The script shows an example how to do this. The property files in `conf/` contain the description how to use the environment variables and java system properties inside the configuration files.

On Windows 32 and Windows 64 SC can be started as Windows service. The directories `bin/win32` and `bin/win64` contain the necessary tools and guide.

On Unix platforms SC can be started as a daemon. The directories `bin/unix` contain the necessary tools and guide.

### 6.2 Stopping SC

The Service connector is stopped with the script `bin/stop-sc.bat` on Windows or `stop-sc.sh` on unix / linux platforms. The script invokes the `sc-console` which in turn sends a *kill* message to the SC.

**Even if `sc-console` can connect to remote SC, stopping SC is possible only from the local node where SC is running.**

### 6.3 Other console operations

The `sc-console` can be used to manage services and to get information about the SC. The syntax is:

```
java -jar sc-console.jar -h <host> -p <port>
{{{enable|disable|state|sessions}?serviceName=<serviceName>}|
 {inspectCache?serviceName=<serviceName>&cacheId=<cacheId>}|
 {clearCache|dump|kill}}
```

The possible commands are:

```
java -jar sc-console.jar -h localhost -p 7000 enable?serviceName=abc
java -jar sc-console.jar -h localhost -p 7000 disable?serviceName=abc
java -jar sc-console.jar -h localhost -p 7000 state?serviceName=abc
java -jar sc-console.jar -h localhost -p 7000 sessions?serviceName=abc
java -jar sc-console.jar -h localhost -p 7000 inspectCache?serviceName=abc&cacheId=700
java -jar sc-console.jar -h localhost -p 7000 clearCache
java -jar sc-console.jar -h localhost -p 7000 dump
java -jar sc-console.jar -h localhost -p 7000 kill
```

For `serviceName` value regular expression can be used. E.g  
`state?serviceName=.*`

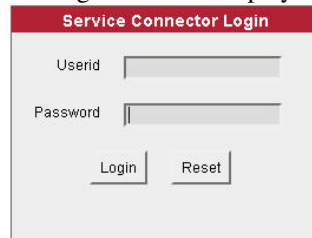
Will display state of all services.

## 6.4 Monitoring SC with the Web GUI

By default the mgmt-port is enabled and configured on port 81. Entering the URL <http://localhost:81/> in your browser will connect with the SC.

### Login

The login screen is displayed when a user connects the Browser to SC.

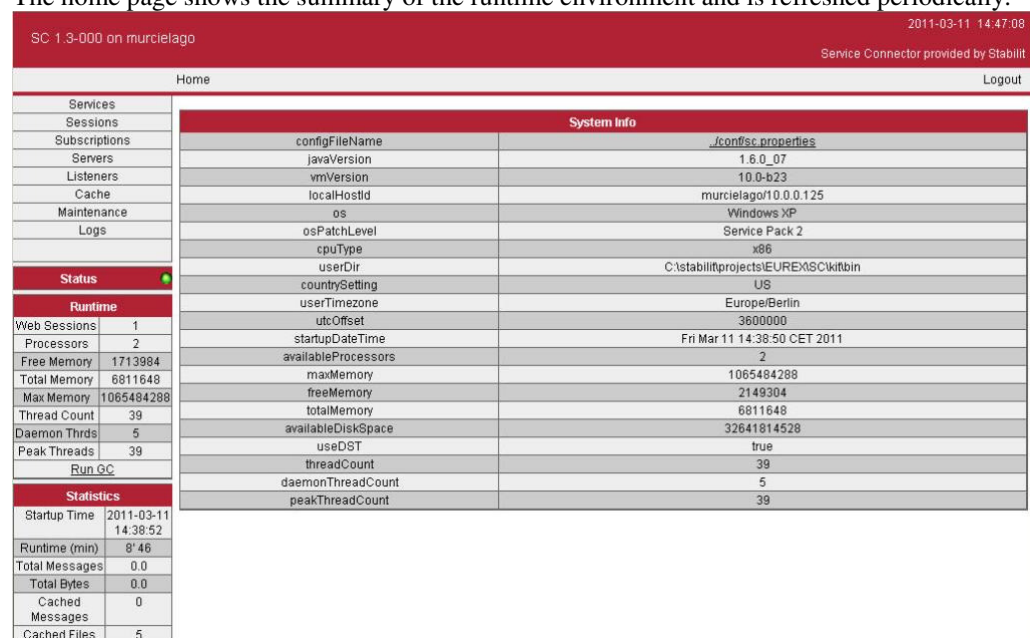


The login form has a red header 'Service Connector Login'. It contains two input fields: 'Userid' and 'Password'. Below the 'Password' field are two buttons: 'Login' and 'Reset'.

The credentials are defined in the configuration file.

### Home

The home page shows the summary of the runtime environment and is refreshed periodically.



The home page has a red header bar with 'SC 1.3-000 on murcielago' on the left and '2011-03-11 14:47:08' on the right. Below the header is a navigation bar with 'Home' and 'Logout'. On the left is a sidebar menu with items: Services, Sessions, Subscriptions, Servers, Listeners, Cache, Maintenance, and Logs. The main content area is divided into three sections: 'Status' (with a green dot), 'Runtime' (showing metrics like Web Sessions, Processors, Memory, Thread Count, etc.), and 'Statistics' (showing Startup Time, Runtime, Messages, Bytes, etc.). On the right is a 'System Info' table with various system parameters.

Parameter	Value
configFileName	./conf/sc.properties
javaVersion	1.6.0_07
vmVersion	10.0-b23
localHostId	murcielago/10.0.0.125
os	Windows XP
osPatchLevel	Service Pack 2
cpuType	x86
userDir	C:\stabilif\projects\leurex\SC\kit\bin
countrySetting	US
userTimezone	Europe/Berlin
utcOffset	3600000
startupDateTime	Fri Mar 11 14:38:50 CET 2011
availableProcessors	2
maxMemory	1065484288
freeMemory	2149304
totalMemory	6811648
availableDiskSpace	32641814528
useDST	true
threadCount	39
daemonThreadCount	5
peakThreadCount	39

The display has several sections:

- The header shows the actual version number and the node name on which SC is running. Actual menu is shown in the navigation line.
- There is a menu structure in the left column
- The status show if SC is running or not. When the GUI loses connection to the SC the green point will be red.
- The runtime box shows summary of the most important runtime parameters.
- The statistic shows an overall SC statistic.
- The System Info areas shows information related to the runtime environment. The link on the configuration file will display its content.

### Services

The service form shows all configured services.

List of services						
Service State	Service Name	Service Type	Servers	Message Queue	Allocated Sessions / Subscriptions	Available Sessions
Enabled	file-1	FILE_SERVICE	-	-	-	-
Enabled	my-session-1	SESSION_SERVICE	0	-	0	0
Enabled	publish-1	PUBLISH_SERVICE	0	0	0	0
Enabled	publish-2	PUBLISH_SERVICE	0	0	0	0
Enabled	sc1-publish-1	CASCADED_PUBLISH_SERVICE	-	0	-	-
Enabled	sc1-session-1	CASCADED_SESSION_SERVICE	-	-	-	-
Enabled	sc2-publish-2	CASCADED_PUBLISH_SERVICE	-	0	-	-
Enabled	sc2-session-2	CASCADED_SESSION_SERVICE	-	-	-	-
Enabled	session-1	SESSION_SERVICE	0	-	0	0
Enabled	session-2	SESSION_SERVICE	0	-	0	0

The link on service state allows enabling or disabling the service interactively. When services are active, counters become links to a drill down and will display detailed information on sessions and servers.

### Sessions

The session form shows all active sessions.

List of sessions				
Service	Session ID	IP Adresslist	Session Timeout (s)	Server
<a href="#">session-1</a>	37883714-c2b3-454a-a170-1a3cb825a286	10.0.0.125/127.0.0.1	12.0	<a href="#">localhost9002</a>

The link on service will display service details, the link on server the server details

### Subscriptions

The subscription form shows all active subscriptions.

List of subscriptions						
Service	Subscription ID	Subscription Mask	IP Addresslist	Subscription Timeout (ms)	No Data Interval (s)	Server
<a href="#">publish-1</a>	4a9e3367-ae5-4062-a8a9-24b5b7e92cd0	0000121ABCDEFHIJKLMNO-----X-----	10.0.0.125/127.0.0.1	60000.0	100	<a href="#">localhost9001</a>

The link on service will display service details, the link on server the server details

### Servers

The server form shows all active servers.

List of servers					
Host	Port	Server Key	Server Type	Max Sessions	Max Connections
localhost	9001	sc2	CASCADED_SC	-	20
localhost	7001	sc1	CASCADED_SC	-	20
localhost	9002	session-1_localhost3386	STATEFUL_SERVER	100	10
localhost	80	fileServer	FILE_SERVER	10	20
localhost	9001	publish-1_localhost3421	STATEFUL_SERVER	10	5

### Listeners

The listener form shows all active listeners.

List of listeners					
Host	Port	Name	Connection Type	Interfaces	Remote Node
127.0.0.1	7000	sc-http	netty.http	127.0.0.1 10.0.0.125 10.0.10.1	-
127.0.0.1	81	mgmt-http	netty.web	127.0.0.1 10.0.0.125 10.0.10.1	-
127.0.0.1	9000	sc-tcp	netty.tcp	127.0.0.1 10.0.0.125 10.0.10.1	-
127.0.0.1	8080	http-proxy	netty-proxy.http	127.0.0.1 10.0.0.125 10.0.10.1	<a href="#">apache</a>

The link on remote node will display node details

### Cache

The cache form shows the state of the cache.

Cache manager configuration				
Status	Disk Path	maxElementsInMemory	maxElementsOnDisk	
ENABLED	..lcache	10000	100000	
List of caches				
Service Name	Composite Size	Element Size	Memory Store Size	Disk Store Size
my-session-1	0	0	0	0
sc2-session-2	0	0	0	0
sc1-session-1	0	0	0	0
session-1	0	0	0	0
session-2	0	0	0	0

### Maintenance

The cache form allows performing maintenance actions.

SC Maintenance			
<a href="#">SC Dump</a>	<a href="#">Show SC Dump List</a>	<a href="#">Delete All SC Dumps</a>	<a href="#">Terminate SC</a>
<a href="#">Clear Cache</a>	<a href="#">Reset Translet</a>		
<a href="#">Property File Download - file-1</a>	<a href="#">Logfile Upload - file-1</a>		

## Logs

The cache form shows the logs and allows their view.

Log files << 2011-03-11		
Logger Name	Appender Name	File
CacheLogger	cacheLog	..\\logs\\sc\\cache.log (1033)
ConnectionLogger	connectionLog	..\\logs\\sc\\connection.log (6276)
root	generalLog	..\\logs\\sc\\sc.log (9350)
MessageLogger	messageLog	..\\logs\\sc\\message.log (920)
PerformanceLogger	performanceLog	..\\logs\\sc\\performance.log (0)
SessionLogger	sessionLog	..\\logs\\sc\\session.log (458)
SubscriptionLogger	subscriptionLog	..\\logs\\sc\\subscription.log (886)

The link left of the date allows browsing previous days. The link on the log file will display its content.

## 6.5 Monitoring SC with the J-Console or Jvisualvm

The script example/jconsole.bat and jvisualvm.bat show how to start jconsole or jvisualvm. When SC is running on your desktop, you can directly select the SC from the local process list. If SC is running on remote node you need to know the node, port of the node and will see SC in the remote process list.

The most interesting parameters are:

- Heap Memory Usage
- Number of Threads
- CPU usage
- VM Summary

With jconsole or jvisualvm it is also possible to set the log level of the running SC.



# 7

## Troubleshooting

### 7.1 General

If you encounter a problem, please work systematically through the following checklist:

- 1) Connect to SC with the Web-GUI and check that it is responding. You can also use the console and get state of the services with `state?serviceName=.*` command.
- 2) Look for ERRORS in `sc.log`

If you find suspicious log entries:

- 1) Use the Web-GUI and create a dump
- 2) Upload log files and the dump to the Web Server
- 3) Send the zip-file to [info@stabilit.ch](mailto:info@stabilit.ch) or directly to the responsible person.

*Note:*

**We understand the user frustration in a case of an error. However we need a precise description of the problem. In order to increase the chance to find and correct the error soon, please do:**

- **Keep cool, handle the case without emotions.**
- **Work systematically. Do not jump from one topic to other.**
- **Communicate precisely, use common terminology.**
- **Write down what you have done, later you will not remember the exact steps.**
- **Do not continue if not absolutely necessary. Other errors may subsequently happen.**
- **Do not repeat the case. It will be more difficult to read its trace.**
- **Save all material, logs, notes, screen shots in one place.**

### 7.2 The most often situations

**SC does not start.**

Start SC manually from the command line and see the displayed error.  
If there is lot of output before SC crashes see the error in the log file.

## 8

## Glossary

**Client**

Piece of an application consuming services and initiating actions.

**Server**

Piece of an application providing services to clients.

**Service**

Abstract unit of work provided by the server and delivered to the client in order to implement a specific functionality. SC supports session, publishing and file services.

**Session**

Temporary allocation of a dedicated server to a client. Session ensures information flow between a client and the allocated server. SC supports request/response sessions and subscription sessions.

**Call**

A call represents a pair of a request and response.

**Command**

A command dispatches specific actions on a server according to the incoming request.

**Request**

Data structure created by the Client or SC in order to initiate an information exchange. It may contain one or more messages.

**Response**

Data structure created by the Server or SC in order to deliver the requested information. It may contain one or more messages.

**Message**

Basic transport instrument to exchange information between client, SC and the server. It belongs to a request or to a response.

**Message Part**

Message part is a piece of a large message. Large message is splitted into parts.

**Composite**

Data structure prepared to hold all message parts of a large message

**Registry**

Common list of known objects that ensures their uniqueness, organized in a way to find them easily.

**Requester**

Piece of code in SC or client initiating the information exchange

**Responder**

Piece of code in SC or server delivering the requested information

**Connection**

Network communication between client and SC or SC and the server

**Endpoint**

Network communication part on SC or server

**Service**

The term service is used as synonym for a piece of application code that implements a certain function.

**Client**

The term client designates the application requesting a certain service.

**Client node**

Client node is physical network node (machine) on which the client application resides.

**Server**

The term server designates the application providing a certain service.

**Server node**

Server node is physical network node (machine) on which the server application resides.

**Message payload**

Message payload is part of the transported message containing the application data. SC does not inspect, transform or interpret the payload. It is the responsibility of the communicating partners to define the format, syntax and semantic of the payload.

**SCMP => SC Message Protocol**

Is a publically presented protocol used for communication between client, SC and the server. In the OSI layer model, message transport protocol is immediately above the application protocol layer 7.

**Message transport format**

Format of messages transported messages between client, SC and the server. It consists of a header and a payload.

**RR => Request / Response**

This is basic communication style. The client requests an execution of a (remote) service and gets back the results in form of a response. The execution may be blocking - clients waits for the response, or non-blocking (asynchronous) – the clients does not wait for the response. Sometimes it is also called Remote Procedure Call = RPC.

**SP => Subscribe / Publish**

This is another popular communication style. The client tells the server what he is interested in and gets the data from the server later when a event arise. The execution is always non-blocking (asynchronous) – the client does not wait for the data receipt. This is because the event occurs on the server.

# Index

## M

Message  
Body • 21

Payload • 21

## R

RPC • 21