



Using the SESM Web Services Gateway

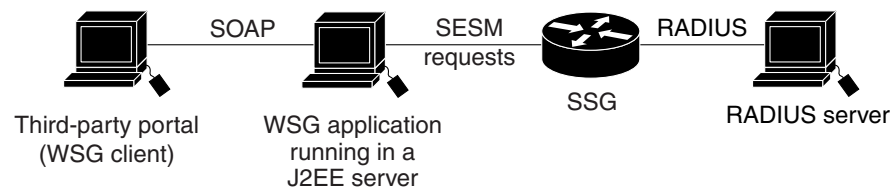
This appendix describes how to install, start, and use the WSG application and sample client. Topics are:

- [Web Services Gateway Introduction, page B-1](#)
- [Installing WSG, page B-2](#)
- [Configuring WSG, page B-2](#)
- [Starting and Stopping the WSG Application, page B-2](#)
- [Running the Demonstration Client Interface, page B-3](#)

Web Services Gateway Introduction

The Web Services Gateway (WSG) application provides a Simple Objects Access Protocol (SOAP)-based interface enabling third-party web portals and subscriber management systems to integrate with the SESM and SSG solution. Any client application can interface with SSG through the WSG using SOAP over HTTP communication. [Figure B-1](#) shows WSG deployment.

Figure B-1 WSG Deployment



The WSG installation includes a web application configured to run in a Jetty container and a command-line client script for demonstration purposes. The WSG web application runs in RADIUS and SPE modes. It does not work in Demo mode.

In this first release, the WSG client interface enables access to the SSG for the following activities:

- Authenticating, starting, and ending sessions on the SSG
- Obtaining session status
- Connecting and disconnecting services

This first release of WSG offers a preview of future development efforts. We invite interested parties to contact us through a Cisco account representative to discuss potential uses for WSG and participate in feature planning efforts for future releases.

Installing WSG

To install WSG:

1. In the SESM installation program, choose the custom installation option.
2. Check the WSG box in the list of custom installation options.

WSG is installed in the `\install_dir\wsg` directory.

Configuring WSG

WSG is configured by default to run in a Jetty container on port 8100.

To change the Jetty container configuration for the WSG application, edit the following file:

```
jetty
  config
    wsg.jetty.xml
```

To change the WSG application configuration, you can either:

- Access the MBeans through the WSG AgentView (port 8200)
- Manually edit the following MBean configuration files:

```
wsg
  config
    lib.xml
    sesm.xml
    dessauth.xml
```

For explanations of the MBeans, see the *Cisco Subscriber Edge Services Manager Web Portal Guide*. The contents of the `sesm.xml` file is the same as the contents of `nwsp.xml`.

Starting and Stopping the WSG Application

The SESM installation process installs and configures the WSG application to run in a Jetty container. To start and stop WSG, run its startup or stop script:

```
jetty
  bin
    WSGstart
    WSGstop
```

This script accepts all of the options and parameters that other SESM web applications use, including the mode option, which allows you to switch between LDAP, RADIUS, and Demo modes at run time. For more information, see the *Cisco Subscriber Edge Services Manager Web Portal Guide*.

The installed default port for the WSG is 8100.

Running the Demonstration Client Interface



Note

The client interface is intended for demonstration purposes only. It can provide an understanding of the WSG interface and possibilities for development. Contact us through your Cisco account representative to discuss your development goals and deployment requirements regarding a WSG interface.

The demonstration client interface script provides command line access to the WSG using SOAP remote procedure calls (RPC). The script is located in:

```
wsg
  bin
    wsgClient
```

To start the client, enter the following command:

```
wsgClient [endpoint]
```

Where *endpoint* is always:

```
http://WSGghost:8100/services/SESM
```

If you do not supply the endpoint, the script provides command usage help. The wsgClient command-line prompt is:

```
wsg>
```

At the prompt, enter **help** to display available commands. At subsequent prompts, enter any of the commands.

Examples

The following examples show the WSG client command-line interface and output from various commands.

```
user1> wsgClient.sh http://localhost:8100/services/SESM
wsg/webapp/WEB-INF/lib/auth.jar:wsg/webapp/WEB-INF/lib/authentication.jar:wsg/webapp/WEB-INF/lib/axis.jar:wsg/webapp/WEB-INF/lib/com.cisco.sesm.contextlib.jar:wsg/webapp/WEB-INF/lib/com.cisco.sesm.lib.jar:wsg/webapp/WEB-INF/lib/com.cisco.sesm.wsg.jar:wsg/webapp/WEB-INF/lib/commons-logging.jar:wsg/webapp/WEB-INF/lib/dess.jar:wsg/webapp/WEB-INF/lib/jaxrpc.jar:wsg/webapp/WEB-INF/lib/jmxri.jar:wsg/webapp/WEB-INF/lib/jmxtools.jar:wsg/webapp/WEB-INF/lib/log4j-1.2.4.jar:wsg/webapp/WEB-INF/lib/mail.jar:wsg/webapp/WEB-INF/lib/protect.jar:wsg/webapp/WEB-INF/lib/saa.jar:wsg/webapp/WEB-INF/lib/sesm.jar:wsg/webapp/WEB-INF/lib/tt-bytecode.jar:wsg/webapp/WEB-INF/lib/wsdl4j.jar:lib/lib/com.cisco.sesm.lib.jar:redis/axis/lib/axis.jar:redis/axis/lib/commons-logging.jar:redis/axis/lib/jaxrpc.jar:redis/axis/lib/log4j-1.2.4.jar:redis/axis/lib/mail.jar:redis/axis/lib/saa.jar:redis/axis/lib/tt-bytecode.jar:redis/axis/lib/wsdl4j.jar:redis/jaxp/lib/crimson.jar:redis/jaxp/lib/jaxp.jar:redis/t/jaxp/lib/xalan.jar
```

```
wsg> help
```

```
act[ivateService] svc [user passwd] - Activate service
auth[enticate] user passwd - Authenticate username/password
dea[ctivate] svc - Deactivate service
end[session] - End the session
get[status] - Get status
h[elp] - This summary
host[key] ip[/port][;name=value ..] - Set hostkey
q[uit] - Quit client
```

```
wsg> hostkey 121.121.122.3
hostkey=121.121.122.3
```

In the preceding command, 121.121.122.3 is the IP address of the subscriber. The SSG must be able to route this address. It uses the address to bind a downlink interface when it creates the edge session for the subscriber.

```
wsg> authenticate ug1-u1 cisco           // username/password respectively
authenticate=true

wsg> get
Identity user: ug1-u1
Service proxy3: off
Service Chris-PT1: off
Service proxy2: ON
Service proxy1: off
Service tunnel7200uk7: off
Service ptSg2: off
Service ptSg1: off
Service pt2: off
Service pt1: off
Service SgMutexSelct1: off
Service sgl: off
Service tunnelNrp4: off
Service Chris-PT-Seq2: off
Service Chris-PT-Seq1: off
Service ?????: off
Service ptMutexSelect3: off
Service ptMutexSelect2: off
Service ptMutexSelect1: off

wsg> act pt1
activate pt1 = true

wsg> get
Identity user: ug1-u1
Service proxy3: off
Service Chris-PT1: off
Service proxy2: ON
Service proxy1: off
Service tunnel7200uk7: off
Service ptSg2: off
Service ptSg1: off
Service pt2: off
Service pt1: ON
Service SgMutexSelct1: off
Service sgl: off
Service tunnelNrp4: off
Service Chris-PT-Seq2: off
Service Chris-PT-Seq1: off
Service ?????: off
Service ptMutexSelect3: off
Service ptMutexSelect2: off
Service ptMutexSelect1: off

wsg> dea pt1
deactivate pt1

wsg> act tunnelNrp4 cisco cisco           // An example of activating authenticated tunnel service
activate tunnelNrp4 = true

wsg> get
Identity user: ug1-u1
Service proxy3: off
Service Chris-PT1: off
Service proxy2: ON
Service proxy1: off
```

```
Service tunnel7200uk7: off
Service ptSg2: off
Service ptSg1: off
Service pt2: off
Service pt1: off
Service SgMutexSelct1: off
Service sg1: off
Service tunnelNrp4: ON
Service Chris-PT-Seq2: off
Service Chris-PT-Seq1: off
Service ?????: off
Service ptMutexSelect3: off
Service ptMutexSelect2: off
Service ptMutexSelect1: off
```

```
wsg> deactivate tunnelNrp4
deactivate tunnelNrp4
```

```
wsg> get
Identity user: ugl-u1
Service proxy3: off
Service Chris-PT1: off
Service proxy2: ON
Service proxy1: off
Service tunnel7200uk7: off
Service ptSg2: off
Service ptSg1: off
Service pt2: off
Service pt1: off
Service SgMutexSelct1: off
Service sg1: off
Service tunnelNrp4: off
Service Chris-PT-Seq2: off
Service Chris-PT-Seq1: off
Service ?????: off
Service ptMutexSelect3: off
Service ptMutexSelect2: off
Service ptMutexSelect1: off
```

```
wsg> end
endSession
```

```
wsg> quit
test-user-u10:165>
```

