# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About StreamStudio</td>
<td>5</td>
</tr>
<tr>
<td>StreamStudio web applications</td>
<td>6</td>
</tr>
<tr>
<td>Component overview</td>
<td>7</td>
</tr>
<tr>
<td>Collector specific components</td>
<td>8</td>
</tr>
<tr>
<td>Composition Center specific components</td>
<td>9</td>
</tr>
<tr>
<td>HTTP web server as front-end proxy</td>
<td>9</td>
</tr>
<tr>
<td>StreamStudio user profiles</td>
<td>10</td>
</tr>
<tr>
<td>StreamStudio communication</td>
<td>12</td>
</tr>
<tr>
<td>StreamStudio security</td>
<td>13</td>
</tr>
<tr>
<td>StreamStudio administrator's tasks</td>
<td>14</td>
</tr>
<tr>
<td>StreamStudio installations</td>
<td>15</td>
</tr>
<tr>
<td>Installation requirements</td>
<td>16</td>
</tr>
<tr>
<td>Recommended JVM memory settings</td>
<td>16</td>
</tr>
<tr>
<td>Running in isolated environments</td>
<td>17</td>
</tr>
<tr>
<td>Advanced StreamStudio configurations</td>
<td>19</td>
</tr>
<tr>
<td>Load-balancing service gateways</td>
<td>20</td>
</tr>
<tr>
<td>Setting up a Trusted Communication Channel</td>
<td>21</td>
</tr>
<tr>
<td>Manually setting up TCC</td>
<td>22</td>
</tr>
<tr>
<td>Changing to PKCS12 keystore (optional)</td>
<td>22</td>
</tr>
<tr>
<td>Creating a keystore</td>
<td>23</td>
</tr>
<tr>
<td>Creating a truststore</td>
<td>23</td>
</tr>
<tr>
<td>Enabling TCC for StreamStudio</td>
<td>24</td>
</tr>
<tr>
<td>Using TLSv1 instead of SSLv3</td>
<td>24</td>
</tr>
<tr>
<td>Creating truststore on UNIX, example</td>
<td>25</td>
</tr>
<tr>
<td>Additional security provider configuration for UNIX</td>
<td>25</td>
</tr>
<tr>
<td>Enabling stronger encryption</td>
<td>26</td>
</tr>
<tr>
<td>TCC Troubleshooting</td>
<td>26</td>
</tr>
<tr>
<td>Managing fonts for Composition Center and Composer</td>
<td>27</td>
</tr>
<tr>
<td>Running StreamStudio</td>
<td>29</td>
</tr>
<tr>
<td>Accessing and logging on to StreamStudio</td>
<td>30</td>
</tr>
<tr>
<td>Accessing StreamStudio</td>
<td>30</td>
</tr>
<tr>
<td>Logging on to StreamStudio</td>
<td>31</td>
</tr>
<tr>
<td>Checking status for StreamStudio</td>
<td>32</td>
</tr>
<tr>
<td>Using the log file</td>
<td>32</td>
</tr>
</tbody>
</table>
About StreamStudio

StreamStudio contains web applications which provide an interface to the StreamServer applications, repositories, and user directories.

For example, via this interface a StreamStudio user can:

• Manage content in business documents by writing personalized texts and adding images.
• Select a business document that was processed by a StreamServer application and re-send this document via a suitable channel, for example email or printer.
• Select distribution method, for example email or fax, for a specific document type sent to a specific user.

About this document

This document gives an overview of the StreamStudio web applications, the involved components, and the tasks to be carried out before using StreamStudio in a business environment. Any tasks and configurations covered in other documentation are clearly referenced to.

In this section

• StreamStudio web applications on page 6.
• Component overview on page 7.
• StreamStudio user profiles on page 10.
• StreamStudio communication on page 12.
• StreamStudio security on page 13.
• StreamStudio administrator’s tasks on page 14.
StreamStudio web applications

The StreamStudio portal includes the following web applications:

- **Composition Center** – For adding messages to output documents. The message can include dynamic texts (for example, customer name) and images based on different conditions.
- **Collector** – For viewing, re-sending, and deleting stored documents. The user can also add annotations to stored documents.
- **Reporter** – For viewing, reprocessing, and deleting jobs.
- **Dispatcher** – For distributing documents via different channels based on user preferences.
- **Composer** – For adding short messages to the documents based on different conditions.
- **Customers** – For administering external users and associating roles with external users.
- **Administrator** – For administering internal users and associating roles with internal users.

For information on how to use the StreamStudio web applications, see the *StreamStudio* online help.

![Figure 1 StreamStudio web portal](image-url)
Component overview

This section gives an overview of the basic components used when running StreamStudio. For detailed description of each component and for a complete overview of all components involved when running the StreamServe software, see the Control Center documentation.

Figure 2  StreamStudio components

StreamStudio web portal
The StreamStudio web portal contains the StreamStudio web applications. The web applications are based on Java code and must be deployed to a Java application server, for example Apache Tomcat.

User directories
The user directories contain user profiles. The user profiles are used to access the StreamStudio web applications via the web portal. The user directories are connected to directory servers, which must support the LDAP protocol.

Note: You can also access StreamStudio without a user directory. For more information, see StreamStudio user profiles on page 10.

Runtime repository
The runtime repository stores jobs and job related information. The repository also stores security profiles and web access information for the StreamStudio web applications.

The runtime repository includes the following repositories:

- **Queue** – Stores jobs and job information in queues.
- **Security** – Stores security settings, for example StreamStudio roles for the user profiles in the user directory.
- **Web Access** – Stores access information for the roles, for example which web applications and documents each role can access.
**StreamServer application**

The StreamServer application runs an exported Design Center Project. Each Design Center Project requires a separate StreamServer application.

**Service gateway**

The service gateway is used to access the user directories, the runtime repository and the StreamServe archive from StreamStudio.

When a user logs on to StreamStudio, the service gateway authenticates the user credentials in the user directory and in the Security repository. The service gateway then gives the user access to certain web applications and documents as defined in the Web Access repository.

**Note:** Some web applications, for example Administrator and Reporter, communicate directly with the runtime repository, without passing the service gateway. However, to enable log on to StreamStudio, a service gateway is still required.

**Collector specific components**

To run StreamStudio Collector, you need a StreamServe archive and an Archiver application in addition to the basic StreamStudio components.

![StreamStudio Collector components](image)

**StreamServe archive**

The StreamServe archive stores output documents and related metadata that are accessed from the Collector web application. When a user searches for documents in the StreamServe archive, all requests and responses are sent through the service gateway.

When a user stores searches, these are stored on the runtime repository.

**Archiver application**

The Archiver application transfers output documents and metadata from the runtime repository to the StreamServe archive according to a specified schedule.
Composition Center specific components

To run StreamStudio Composition Center, you need a web content repository in addition to the basic StreamStudio components.

Figure 4 StreamStudio Composition Center components

The web content repository is used by the Composition Center web application for storing document definitions, resources, and rules during the document design phase.

When a document definition is approved, the document definition together with its resources and rules is copied from the web content repository to the runtime repository, where it is available to the StreamServer application that produces the document.

HTTP web server as front-end proxy

You can use an HTTP web server as a front-end proxy between the clients and the Java application server on which StreamStudio runs.

For example, if you have many concurrent users, the web server may enhance performance. The web server can also be used to provide static content, such as illustrations, which it normally does faster than the Java application server.

Figure 5 Client(s) and web server when running StreamStudio
StreamStudio user profiles

StreamStudio user profiles are used to access the StreamStudio web applications via the StreamStudio web portal.

The StreamStudio user profiles are stored in user directories. However, you can also access StreamStudio without a user directory. For example, if the company is small and does not have a user directory, or if you want to demonstrate StreamStudio without a directory connection. In this case, the only available user is the application domain administrator. See Application domain administrator with System Manager role on page 11.

User directories

Most often, the StreamStudio user profiles are stored in user directories, connected to directory servers. The directory servers must support the LDAP protocol. You can either use Microsoft Active Directory or a generic LDAPv3 compliant directory, for example OpenDS.

Internal and external users

The users are either internal users (for example, company employees), or external users (for example, company customers). StreamStudio can authenticate users from a single user directory (with internal and external users in separate locations) or from multiple user directories (with internal and external users in different user directories).

You configure the user directories for the internal and external users in Control Center. For more information, see the Control Center documentation.

Associating users with roles

You must associate the user profiles with roles, for example administrator, call center personnel, etc. Depending on role, the users have different access rights to the StreamStudio web applications and to stored documents.

For internal users, you define the roles using the Administrator web application. For external users, you define the roles in the Customers web application. For more information, see the StreamStudio online help.

Defining access for roles

Each role has either no access, full access, or basic access to the StreamStudio web applications. The default basic accesses to the web applications are listed in the StreamStudio online help.

You can modify the basic accesses using the Administrator and Customers web applications. For example, you can define a role that only has access to the Collector web application and to invoices sent to certain customers. For more information, see the StreamStudio online help.
**Application domain administrator with System Manager role**

To define roles and accesses, you must log on to the StreamStudio web portal.

The first time you log on, you must log on as the StreamServe application domain administrator. This administrator is defined in Control Center, and is by default assigned the System Manager role in StreamStudio. For more information, see the *Control Center* documentation.

When logged on as the application domain administrator, you can define new roles. For example, you can define a role with StreamStudio System Manager rights and assign this role to a user in the user directory.
StreamStudio communication

StreamStudio uses web services (WS) and JDBC when communicating with the service gateway, the runtime repository, and the web content repository. The service gateway, in turn, uses ODBC when retrieving data from the runtime repository and from the StreamServe archive.

![Diagram of StreamStudio communication]

**Figure 6  StreamStudio communication**

**Web services**

Web services are used when StreamStudio communicates with the service gateway.

You configure the service gateway web services in Control Center. Some additional configurations can be done in the property file for the web services (ws.properties). For example, in this file you can load balance two service gateways, or specify settings for a Trusted Communication Channel (TCC). See *Advanced StreamStudio configurations* on page 19.

**JDBC**

StreamStudio uses JDBC stored procedures when retrieving data directly from the runtime repository and the web content repository. The required drivers (JTDS driver for Microsoft SQL Server and DataDirect JDBC for Oracle) are provided by the StreamStudio setup.

The property files for SQL Server (sql_jdbc.properties) and Oracle (oracle_jdbc.properties) are pre-configured and installed together with the StreamStudio setup. Normally, you do not have to edit these files. However, if you set up StreamServe for Oracle Real Application Clusters (RAC), you must generate a new oracle_jdbc.properties. (To set up StreamServe for Oracle RAC, there are also some other configurations that must be carried out.)
StreamStudio security

There are several ways in which you can enhance the security of the StreamStudio installation. For example, you can use web service security and a Trusted Communication Channels (TCC). You can also run in isolated environments with firewalls in between.

Figure 7  Example, web service security (WSS), TCC, and firewall

Web service security

In Control Center, you can specify that all web service requests from StreamStudio to the service gateway are authenticated with the credentials of the user logged in to StreamStudio. For more information, see the Control Center documentation.

Trusted Communication Channel (TCC)

If you use web service security, the user credentials and requests are sent in clear text in the web service requests. To further enhance security, it is recommended to encrypt the communication between StreamStudio to the service gateway by configuring a Trusted Communication Channel (TCC). For more information, see Setting up a Trusted Communication Channel on page 21.

Firewalls

You can also enhance security by using isolated environments with firewalls in between. The company environment decides where to put the firewalls. For example:

- If a company has an existing front-end web server, you can use a separate computer for the Java application server (on which StreamStudio runs) and install a firewall in between.

- If a company exposes the web services, a firewall between the Java application server and the service gateway may be suitable.

Note: If you combine a firewall between the Java application server and the service gateway with a TCC, you must make sure the SSL packages are sent without parsing.
StreamStudio administrator’s tasks

This section gives an overview of the StreamStudio specific tasks that must be carried out before StreamStudio can be used in a business environment.

StreamStudio tasks

1. Install StreamStudio. See StreamStudio installations on page 15.
2. Carry out the required Project preparations in Design Center. See:
   - Composition Center documentation.
   - Collector documentation.
   - Reporter documentation.
   - Dispatcher documentation.
   - Composer documentation.
3. Configure the StreamStudio environment in Control Center. For example, configure the application domain for StreamStudio, deploy the StreamStudio web archive file to the Java application server, etc. See the Control Center documentation.
5. Run StreamStudio (access and log on to StreamStudio, check status of involved services, etc.). See Running StreamStudio on page 29.
6. Associate the users in the user directory with roles and configure accesses for the roles. See the StreamStudio online help.

Running in isolated environments

If Control Center and Framework is installed on a different computer than the Java application server, you can still configure the StreamStudio environment in Control Center. Then you must manually copy the web archive file and the web portal configuration file to the computer where the Java application server is installed.

See Running in isolated environments on page 17.
StreamStudio installations

The StreamStudio web applications are based on Java code and must be deployed to a Java application server. The application server executes distributed application code, such as Java beans and servlets.

**Deploy StreamStudio from Control Center**

You deploy the StreamStudio web portal to the Java application server from Control Center. In Control Center, you specify the directory (that is, the portal root) on the Java application server to which you want to deploy the StreamStudio web archive file. The management gateway copies the file to the specified portal root from where the file can be unpacked. For more information, see the Control Center documentation.

**Deploy several StreamStudio web portals**

Several StreamStudio web portals can be deployed to the same Java application server. This enables you to run StreamStudio Persuasion SP4 and later Persuasion versions on the same application server. The earliest release must be installed first.

Deploying several StreamStudio web portals to the same Java application server is resource intensive (i.e. memory) and may result in decreased performance.

For memory allocation recommendations, see Recommended JVM memory settings on page 16.

**Deploy manually in isolated environments**

A common production scenario is that Framework and Control Center is installed on a different computer than the Java application server. In such a scenario, you can still configure the StreamStudio environment in Control Center. Then you must manually copy the web archive file and the web portal configuration file to the computer where the Java application server is installed. See Running in isolated environments on page 17.

**In this section**

- Installation requirements on page 16.
- Running in isolated environments on page 17.
Installation requirements

To run StreamStudio, you must install the following:

• Framework and Control Center.
• StreamStudio.
• A Java application server.
• Java Runtime Environment (included in the Framework and Control Center setup). The Java Runtime Environment must be installed on the same computer as the Java application server.
• A web browser and PDF reader (for Composition Center preview).

You can also install the following:

• A directory server (optional). If no directory server is installed, the only available StreamStudio user is the application domain administrator.
• An HTTP front-end web server (optional).
• Firewalls (optional).

Related topics

• For information on how to install the StreamStudio software and Framework and Control Center, see the Installation Guide.
• For supported software and versions, see the Supported platforms and software documentation.

Recommended JVM memory settings

You must make sure there is enough memory allocated for StreamStudio on the JVM (Java Virtual Machine). How much memory required depends on the current deployment scenario. For example, deploying several StreamStudio web portals to the same Java application server requires more memory than deploying a single web portal.

As a rule of thumb, StreamServe recommends the following memory settings when deploying a single StreamStudio web portal to a Java application server:

- \( \text{-Xms128m} \)
- \( \text{-Xmx512m} \)
- \( \text{-XX:PermSize=256m} \)
- \( \text{-XX:MaxPermSize=512m} \)

For information on how to specify the memory settings, see the user guide for the Java application server.
Running in isolated environments

A common production scenario is that Framework and Control Center is installed on a different computer than the Java application server, with a firewall in between. Since the Java application server computer has no management gateway installed, you cannot access the configuration file for the StreamStudio web portal or deploy the StreamStudio web archive file.

To solve this, you configure the StreamStudio environment in Control Center. Then you must manually copy the web archive file to the computer where the Java application server is installed. Finally you copy the web portal configuration file to the Java application server computer.

To copy and deploy the StreamStudio Web Archive file

1. Configure the StreamStudio environment in Control Center. See the Control Center documentation.
   
   **Note:** When you configure StreamStudio web portal in Control Center, you must select the Manual deploy option.

2. Copy the `<Portal name>.war` file from following directory on the Control Center computer:
   
   `<Base Directory>\<Version>\root\etc\unmanagedportals`

   Where `<Base directory>` is the path specified for StreamServe Projects during the Framework and Control Center installation.

   For example: `C:\ManagementGateway`

3. Paste the file into the following directory on the Java application server computer:
   
   `<Portal root>`

   Where `<Portal root>` is the directory for the web archive file.

   For example, for Apache Tomcat: `<TOMCAT_HOME>\webapps\`
Running in isolated environments
StreamStudio installations

4 Unpack the web application file. See the user documentation for the Java application server. For Apache Tomcat, the file may be unpacked automatically if you restart the Apache Tomcat service.

To copy the StreamStudio configuration file

1 Copy the domains folder, containing the web portal configuration file, from the following directory on the Control Center computer:
   `<Base Directory>\<Version>\root\etc\unmanagedportals\<Portal name>\`

   **Note:** Do not copy the complete `<Portal name>` folder, since this catalogue is created by the Java application server when the web application file is unpacked.

2 Paste the domains folder into the following directory on the Java application server computer:
   `<Portal root>\<Portal name>`

   Where:
   - `<Portal root>` — Is the directory for the web archive file. For example, for Apache Tomcat: `<TOMCAT_HOME>\webapps`
   - `<Portal name>` — Is the web portal name as specified in Control Center.
Advanced StreamStudio configurations

This section contains advanced StreamStudio configurations, which cannot be configured in Control Center. For example:

- If you want to load balance the primary and secondary service gateways specified in Control Center, you must edit the property file \( \text{ws.properties} \) for the web services.
- If you want to encrypt the communication between the Java application server (on which StreamStudio runs) and the service gateway, you must set up a Trusted Communication Channel (TCC).
- To add fonts and font sizes to the text editor used in Composition Center and Composer, you must edit the \( \text{mcesettings.xml} \) file.

In this section
- Load-balancing service gateways on page 20.
- Setting up a Trusted Communication Channel on page 21.
- Managing fonts for Composition Center and Composer on page 27.
Load-balancing service gateways

In Control Center, you can specify a primary and a secondary service gateway. The secondary service gateway is used only if the primary service gateway becomes unavailable. Instead of this default behavior, you can load balance the gateways. The Java application server then connects to the service gateway that was used at the last occasion. This is suitable if you have two or more Java application servers hosting StreamStudio and want to minimize the load on one specific service gateway.

You can also modify the following timeout values:

- The time that a failed service gateway will be unavailable for connection retries if there are two or more available service gateways. The default value is 300 000 ms (5 minutes).
- The time that a failed service gateway will be unavailable for connection retries if there is only one or no other available service gateway. The default value is 500 ms.

To configure load balancing of service gateways

1. Open the `ws.properties` file, located in the following directory:
   `<Portal root>\<Portal name>\WEB-INF\spring\properties`
   For example, for Apache Tomcat:
   `<TOMCAT_HOME>\webapps\<Portal name>\WEB-INF\spring\properties`

2. Change the `ws.servicegateway.url.lb.roundrobin` property to `true`. When set to `false`, the service gateways act in the default primary and secondary gateway configuration.

To modify the service gateway timeout values

1. Open the `ws.properties` file, located in the following directory:
   `<Portal root>\<Portal name>\WEB-INF\spring\properties`
   For example, for Apache Tomcat:
   `<TOMCAT_HOME>\webapps\<Portal name>\WEB-INF\spring\properties`

2. Modify the following values:

   - `ws.servicegateway.bad.reputation.long=300000`
   - `ws.servicegateway.bad.reputation.short=500`
Setting up a Trusted Communication Channel

During the Framework and Control Center setup you can specify Advanced Security which configures the service gateway(s) to use a Trusted Communication Channel (TCC) between the service gateway(s) and StreamStudio.

The setup only configures TCC on the service gateway side. This means you must manually configure TCC on the Java application server side. This section describes how to set up TCC for StreamStudio on an Apache Tomcat application server. Corresponding procedures can be performed on SAP NetWeaver or IBM WebSphere.

Due to the US export policy, a weaker encryption has to be the default. To use a stronger encryption, see Enabling stronger encryption on page 26.

Note: To set up a TCC manually, you should have good knowledge in PKI, Apache Tomcat, and StreamStudio.

After all configurations are done, restart Apache Tomcat.

To set up a TCC, you need a keystore and a truststore, which requires a keytool utility. For information on the keytool utility, see http://java.sun.com/j2se/1.3/docs/tooldocs/win32/keytool.html

Prerequisites

• Framework and Control Center is installed with Advanced Security configured.

• StreamStudio is installed.

• The StreamServer(s) used with StreamStudio applications must be enabled for TCC and be using SSLv3.

• Check that the TOMCAT_HOME environment variable is set to the path of the Tomcat installation StreamStudio is running on.

• J2SE 5.0 Runtime Environment is installed.

• A valid CA certificate must be available, for example strs_ca.cert

• A CA signed certificate that authenticates StreamStudio on the StreamServer(s) must be available, for example streamstudio.cert

• If you want to use PKCS12, a PKCS12 keystore called keystore.p12 that contains streamstudio.cert and streamstudio.pem must be available.

• A CA signed certificate that authenticates the StreamServer(s) on StreamStudio must be available, for example trustedserver.cert.
In this section

• Manually setting up TCC on page 22.
• Enabling stronger encryption on page 26.
• TCC Troubleshooting on page 26.

Manually setting up TCC

Note: This describes how to set up TCC for StreamStudio on an Apache Tomcat application server. Corresponding procedures can be performed on SAP NetWeaver or IBM WebSphere.

You must:

• Create a truststore that contains the CA certificate and all trusted server certificates.
• Enable TCC for StreamStudio.

If you want to use a PKCS12 keystore instead of the default JKS keystore, you must modify ws.properties to use a PKCS12 keystore. See Changing to PKCS12 keystore (optional) on page 22.

If you use JKS (default), you must create a keystore. See Creating a keystore on page 23.

In this section

• Changing to PKCS12 keystore (optional) on page 22.
• Creating a keystore on page 23.
• Creating a truststore on page 23.
• Enabling TCC for StreamStudio on page 24.
• Using TLSv1 instead of SSLv3 on page 24.
• Creating truststore on UNIX, example on page 25.
• Additional security provider configuration for UNIX on page 25.

Changing to PKCS12 keystore (optional)

1. Create a \certs folder in the following directory:
   \TOMCAT_HOME\webapps\<Portal name>\WEB-INF\certs
2. Copy keystore.p12 to
   \TOMCAT_HOME\webapps\<Portal name>\WEB-INF\certs
3. In ws.properties, change the ws.https.keystore.type type from JKS to PKCS12

   Note: Do not change the property set for trustStore.
Creating a keystore

**Note:** Tomcat usually manages to read a .p12 or .pfx file directly. This means that you do not need to create a keystore, only the truststore. However, if JKS is used, you must create a keystore.

The following is an example of creating a keystore for Tomcat on Windows. Corresponding steps are performed on UNIX.

**Note:** The keystore must contain the private key for Tomcat.

**To create a keystore**

1. Browse to `<TOMCAT_HOME>/webapps/<Portal name>/WEB-INF/certs`

2. Run the following command:
   ```
   keytool -import -file "D:\Certificates\tomcat.streamserve.com.cer" -keystore "C:\Program files\Apache Software Foundation\Tomcat 5.0\webapps\<Portal name>\WEB-INF\certs\keystore.p12"
   ```

3. Enter keystore password
   
   password0
   
   Check the information listed and select whether you trust the certificate. If trusted, the certificate is added to the keystore.

Creating a truststore

**Note:** For an example of how to create a truststore for Tomcat on UNIX, see *Creating truststore on UNIX, example* on page 25.

1. Run the following command from the `\certs` folder:
   ```
   keytool -import -trustcacerts -alias "<valid_CA_certificate>" -file <valid_CA_certificate>.cert -keystore <TOMCAT_HOME>/webapps/<Portal name>/WEB-INF/certs\truststore.jks -storepass "<truststore_password>"
   ```

2. For each StreamServer that should be trusted by StreamStudio, run the following command:
   ```
   keytool -import -keystore <TOMCAT_HOME>/webapps/<Portal name>/WEB-INF/certs \truststore.jks -storepass "<truststore_password>" -file <CA_signed_certificate>.cert
   ```
Enabling TCC for StreamStudio

1. Un-comment the following line in `<TOMCAT_HOME>/webapps/<Portal name>/WEB-INF/spring/spring.xml`
   `<import resource="ws-security.xml"/>

2. In `<TOMCAT_HOME>/webapps/<Portal name>/WEB-INF/spring/properties/ws.properties`, set the property values according to the following table:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ws.servicegateway.url</td>
<td>https://&lt;certificate URL address&gt;:2718</td>
</tr>
<tr>
<td>ws.https.keystore.url</td>
<td>file:&lt;TOMCAT_HOME&gt;/webapps/&lt;Portal name&gt;/WEB-INF/certs/keystore.p12</td>
</tr>
<tr>
<td>ws.https.keystore.password</td>
<td>&lt;keystore_password&gt;</td>
</tr>
<tr>
<td>ws.https.truststore.url</td>
<td>file:&lt;TOMCAT_HOME&gt;/webapps/&lt;Portal name&gt;/WEB-INF/certs/truststore.jks</td>
</tr>
<tr>
<td>ws.https.truststore.password</td>
<td>&lt;truststore_password&gt;</td>
</tr>
<tr>
<td>ws.https.ssl.protocol</td>
<td>SSLv3</td>
</tr>
</tbody>
</table>

3. Check the territory.xml in `<TOMCAT_HOME>/webapps/<Portal name>/WEB-INF/spring/properties` that the service gateway’s URIs use https, not http.

Using TLSv1 instead of SSLv3

**Note:** The StreamServer(s) must be TLSv1 enabled. Check in `<TOMCAT_HOME>/webapps/<Portal name>/WEB-INF/spring/properties/ws.properties`

- In `<TOMCAT_HOME>/webapps/<Portal name>/WEB-INF/spring/properties/ws.properties`, set the property values according to the following table

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ws.https.ssl.protocol</td>
<td>TLSv1</td>
</tr>
</tbody>
</table>
Creating truststore on UNIX, example

1. Run the following command:
   ```
   keytool -import -trustcacerts -alias "strs_ca" -file /nfshome/user/strs/sol/ca.crt -keystore.truststore.jks
   ```
2. Enter password
   ```
   password0
   ```
3. Check the info displayed.
4. Select to trust the certificate.
5. Run the following command:
   ```
   keytool -import -keystore <TOMCAT HOME>/webapps/<Portal name>/WEB-INF/certs/truststore.jks -storepass "<TRUST_PASSWORD>"
   -file trustedserver.cert
   ```
6. Add a trusted public key by running the following command:
   ```
   keytool -import -keystore truststore.jks -file /nfshome/user/machine.streamserve.com.crt
   ```
7. Enter keystore password:
   ```
   password0
   ```
   The Certificate is added to the keystore.

Additional security provider configuration for UNIX

On UNIX, you must use the vendor specific JCE implementation. For example, you must use Sun’s implementation if you use Sun’s Java Virtual Machine (JVM), and IBM’s implementation if you use IBM’s JVM.

On some JVM distributions, the BouncyCastle JCE implementation is used by default.

Check that, for example, Sun’s JCE implementation is placed before the BouncyCastle implementation in the security provider list found in `/opt/jre1.5.0_15/lib/security/java.security`. If not, insert the following, e.g.:
```
security.provider.X=com.sun.crypto.provider.SunJCE
security.provider.X+1=org.bouncycastle.jce.provider.BouncyCastleProvider
```

**Note:** Do not break the sequence, there must be an ordered list of security providers.
Enabling stronger encryption

Download and install the following files:

- US_export_policy.jar
- local_policy.jar

You can find them in [http://www.sun.com/download/index.jsp](http://www.sun.com/download/index.jsp)

Search for "Java(TM) Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 5.0"

To install policy files

1. Copy the files to JRE_HOME, for example /opt/jre1.5.0_15/lib_security
2. Restart Tomcat.

TCC Troubleshooting

- Check the <TOMCAT_HOME>/logs/streamstudio.log
- Run in debug mode by setting log4j.logger.com.streamserve=DEBUG in <TOMCAT_HOME>/webapps/<Portal name>/WEB-INF/log4j.properties
Managing fonts for Composition Center and Composer

To add fonts and font sizes to the text editor used in Composition Center and Composer, you edit the mcesettings.xml file. The mcesettings.xml file is located in (Apache Tomcat example):

<TOMCAT_HOME>/webapps/<Portal name>/WEB-INF/settings

Composer specifics

• Arial must always be available to the StreamServer application, since it is used when no font is selected for a text.

• If a font is not available to the StreamServer application, texts with that font are not displayed in output documents.

Composition Center specifics

• Times New Roman must always be available to the StreamServer application. This font is used when no font is selected for a text.

To add a font and specify the font sizes

1 Open the mcesettings.xml file in a text editor.

2 In the theme_advanced_fonts setting, add the font with the following syntax:

<option value="font_name">font_display_name</option>

where font_name is the name of a font and font_display_name is displayed in the drop-down list in the editor.

3 In the theme_advanced_fontsize setting, add the non-standard font sizes with the following syntax:

<option value="font_size">font_display_size</option>

Where font_size is the font size and font_display_size is displayed in the drop-down list in the editor.

To restart the Java application server

To implement the changes made in the mcesettings.xml file, you must restart the Java application server.

To make fonts available to the StreamServer application

All fonts used in Composition Center and Composer must be available to the StreamServer application. In the Design Center Project configuration, import the required fonts to a resource set connected to the Platform.
Running StreamStudio

This section covers how to access the StreamStudio web portal and how to check the services StreamStudio is depending upon. For trouble-shooting, you can view the StreamStudio log file.

In this section
• Accessing and logging on to StreamStudio on page 30.
• Checking status for StreamStudio on page 32.
• Using the log file on page 32.
Accessing and logging on to StreamStudio

You access StreamStudio directly in the web browser.

In the user directories, StreamStudio users are associated with application domains. A user can only access the application domains that the user is associated with. When logging on to StreamStudio, the user enters user name and password, and selects an application domain. In the list of available application domains, all domains that the StreamStudio web portal is connected to are displayed (that is, not only the ones that the specific user is associated with).

To facilitate the login, you can add the application domain to the URL used when starting StreamStudio. The user then enters user name and password and is automatically directed to the specified domain. See Accessing StreamStudio below.

Single sign-on

StreamStudio has support for single sign-on, that is it can handle a remote secure token (that is, a REMOTE_USER). For more information, see the user documentation for the front-end web server.

In this section

• Accessing StreamStudio on page 30.
• Logging on to StreamStudio on page 31.

Accessing StreamStudio

You access StreamStudio directly in the web browser, using the following URL:

http://<host>:<port>/<Portal name>/Portal/start

Where:

• <host> – Is the server that runs the Java application server.
• <port> – Is the port that the Java application server listens to (by default, 8080).

Note: If you have a front-end HTTP server on port 80, you do not have to specify the port number.

• <Portal name> – Is the name of the web portal as specified in Control Center.

To direct the user to a certain domain, add the following to the syntax above:

?domain=<Application domain>

Where:

• <Application domain> – Is the name of the application domain that the user should be directed to.

For example:

http://localhost:8080/MyWebPortal/Portal/start?domain=Production
Logging on to StreamStudio

To be able to log on to StreamStudio, a user must be assigned membership to a role.

By default, the application domain administrator is assigned the System Manager role in StreamStudio and has full access to the web applications. The first time you log on to StreamStudio, you must log on as the application domain administrator. You can then define other roles and accesses using the Administrator and Customers web applications and assign StreamStudio users membership to the roles.

The application domain administrator is configured in Control Center, see the *Control Center* documentation. The administrator can be used for StreamStudio log on, regardless of user directory vendor or if no user directory is available.

---

**Default StreamServe users for test and demonstration**

If you install a user directory for test or demonstration purposes, you can populate the directory with the following default users:

- **User name:** strsAdmin  
  **Password:** insecure

- **User name:** strsReader  
  **Password:** insecure

These users are installed during the StreamServe Framework and Control Center setup and must be imported to the user directory during the directory server installation. For more information, see the *Installation Guide > Installing OpenDS*.

Before you can log on with one of these default users, you must assign the user membership to a role.
Checking status for StreamStudio

To check that the service gateway is running in your application domain
1. In Control Center, right-click the service gateway application in your application domain and select Refresh.
2. Check that the State property is Running.

Services that must be started to access StreamStudio
If you cannot start StreamStudio, check that the following services are started:
• StreamServe Management Gateway.
• StreamServe Management Nanny.
• StreamServe Service Gateway.
• The service for the directory server.
• The service for the Java application server. For example, Apache Tomcat.

Using the log file
StreamStudio generates a log file (streamstudio.log) containing information about the web applications. The file is located in the following directory:
<Java_application_server_home>/logs
For example, for Apache Tomcat:
<TOMCAT_HOME>/logs

Log levels
The StreamStudio log has four different levels for the severity of the content:
• ERROR — Displays severe errors.
• WARN — Displays non-fatal errors that might need looking into.
• INFO — Displays informational messages.
• DEBUG — Mainly for developers.

The settings can be changed in the log4j.properties file, located in the following directory:
<Portal root>/<Portal name>/WEB-INF
For example, for Apache Tomcat:
<TOMCAT_HOME>/webapps/<Portal name>/WEB-INF

After changing these settings, you must restart the Java application server.