WEBMETHODS CLOUDSTREAMS FAQ AND TROUBLESHOOTING GUIDE

October 2020

CONTENTS

1 Software AG Designer 3
2 CloudStreams Connector 4
3 CloudStreams Connection 5
3.1 Connectivity 6
4 CloudStreams Server 11
5 CloudStreams Governance 13
6 CloudStreams Services 14
7 CloudStreams Connector Listener 15
8 Building CloudStreams assets using webMethods Asset Build Environment 17
9 Pre-requisites for Diagnostics 18
10 Other Considerations 20

1 Software AG Designer

- **Unrecognized node error in Software AG Designer.**

  Client side CloudStreams plugins maybe missing. Check whether *CloudStreams Service Development* is installed in Software AG Designer by navigating to **Help > About Software AG Designer > Installation details** and search for the *webMethods Service Development extension for CloudStreams, com.softwareag.is.ui.cloudstreams.feature.feature.group*. If not available, install it from **Software AG Designer > CloudStreams > CloudStreams Development**.

- **The Connector Service signature is blank or appears as not configured.**

  Review the error message in the pop-up window. This could be due to missing doctypes or maybe the connector descriptor file is modified incorrectly. Also look for detailed errors or warnings in the Software AG Designer error logs.

- **I do not see CloudStreams connector packages, for example, wmSalesforceConnector_v* in Software AG Designer whereas I can see those connector packages in Integration Server.**

  Connector and other packages are packages generated after you install the CloudStreams provider. By default, these packages are not visible in Software AG Designer for performance reasons. These can be made visible in the Integration Server Administration UI by navigating to the CloudStreams solution page. Go to **CloudStreams > Providers > *Provider Name* > Connectors** and click on the *config* link. Then click **Show connector document packages**.

2 CloudStreams Connector

- **What is the extent of JSON support in CloudStreams REST Connector resources?**

  CloudStreams supports most of the variations of the JSON data format, which includes Multi root node, Array root node element with key, Array root node without key for array, Multi array root node with key, and so on.
  Currently, Multi array root node without key is not supported, which means that only one level of anonymity is supported in a Multi array root node without key. This is valid for request and response messages. This scenario can be handled as raw *Streams* instead of *Document Type*.

- **After installation, the latest CloudStreams connectors are not visible.**

  The previously installed or generated connector artifacts may not have cleaned up properly. Ensure that you have applied the latest CloudStreams Server fix. Follow the “Connector Uninstallation” section steps and install the connector again. Always uninstall/delete older provider packages before installing any new version of provider packages.

- **Why do I get the error 'VIRTUAL_SERVICE_INIT_ERROR: Virtual service runtime could not be initialized' while executing a connector service?**

  This error could appear after reloading the custom package. If a custom package has no dependency on the provider package, the artifacts of the package will not be notified to the CloudStreams Server after the custom package reloads. The reason is that a different package classloader will load the package assets and with no dependency on the provider package, the assets will not be identified as meaningful CloudStreams nodes in IS. Hence the connection nodes will not be visible. To fix this, from Software AG Designer, set the custom packages dependency on the provider package and restart Integration Server.

3 CloudStreams Connection

- **We encounter an error invalid_session_id when connecting to a Salesforce.com account during service execution. How can I prevent an invalid session time out?**

  For connectors implementing login/logout sequence, use the Session Management functionality. For Salesforce.com, because we implement the explicit login sequence which refreshes the session, set *Session Management* to "*fixed*" in the Connection Management Properties and set the value to a number lesser than the time out (in mins) set at the Salesforce.com portal. Once *Session Management* is enabled, CloudStreams will ensure that the session does not expire by calling the login sequence behind the scene when it exceeds the time out limit.

- **Why are CloudStreams connections not visible or disappear after reloading the CloudStreams package?**

  If the custom package has no dependency on the provider package, the artifacts of the package will not be notified to CloudStreams Server after the package reloads. The reason is that a different package classloader will load the package assets and with no dependency on the provider package, the assets will not be identified as meaningful CloudStreams nodes in Integration Server. Hence the connection nodes will not be visible to the user. To fix this, from Software AG Designer, set the custom packages dependency on the provider package and restart Integration Server.

- **CloudStreams connections/packages are copied to another Integration Server and the connection is not getting enabled.**

  Due to security concerns, passwords are not part of the connection nodes and stored in the Passman storage. While copying a connection node from one Integration Server to another, it does not copy the connection password. Hence you need to update the connection password explicitly. This is true for all secure fields, for example, OAuth tokens.

- **Why do I get "session expired" or "session not valid" errors?**

  These errors occur because the session has become invalid. Connection becomes invalid once the session expires. To overcome this, depending upon the session management behavior of the back end (provider), enable *Session Management* (fixed/idle/auto) and set appropriate values for *Session Time out* in Connection configuration. This value should be less than the session timeout specified in the back end. The **auto** option is applicable only for OAuth 2.0 back ends, which return the *expires_in* value.

- **OAuth access tokens often expires**

  If you observe that the access tokens of SaaS providers are often expiring, enable *Session Management* and configure the connection for refresh tokens. See the *Administering webMethods CloudStreams User Guide* and relevant connector documentation for more information on the refresh token functionality.

The commonly selected option for refreshing the token is *Body Query String*. You can also write your own custom flow service or Java service to make refresh requests, which adhere to the following specification:
wm.cloudstreams.service.common.lookup.specs:oauthTokenRefreshServiceSpec.
Also ensure that *Session Management* is enabled with a time out value less than the token expiry time in minutes.

- **Socket write errors for AWS S3**

  If you get a socket write error while uploading or downloading a big file from AWS S3, it might be that you are trying to upload to or download from a bucket, which belongs to one of the several non-default regions hosted by AWS. In such cases, set the *Use Expect Continue* property to *true* in Connection configuration.

  To set the *Use Expect Continue* property to *true*, do the following:
  1. Go to **Solutions > CloudStreams**.
  2. Click **Providers**.
  3. Select the connector. All connections under that connector are listed.
  4. Click *Yes* under the *Enabled* column to disable the connection.
  5. Click **Edit**.
  6. Click **Advanced view**.
  7. In the connection configuration properties page, **Transport Protocol** section, set **Use Expect Continue** to **true**.
  8. Re-enable the connection and run the connector service again.

3.1 Connectivity

- **I see the following error while enabling a Salesforce connection**

  "First Element must contain the local name, Envelope, but found html"

  While enabling a Salesforce connection, the handshake process takes place where certificates get exchanged between the server and client.

  In case of a 2-way SSL handshake process, if the server does not send the list of trusted authorities, then by default CloudStreams as a client does not send any certificates to the server.

  In that case, Salesforce sends the following HTML error:

  ```
  <html>
    <head>
      <title>Certificate Error </title>
    </head>
    <body bgcolor=#ffffff
      text=#3198d8>
      <center>
  ```



```
<img
  src="http://www.sfdcstatic.com/common/assets/img/logo-
  company.png">
<p>
<h3>Client
  certificate
  error:<i>unable
  to get
  local
  issuer
  certificate</i>
</h3>
</center>
</body>
</html>
```

To resolve this error, set the following watt property and then restart Integration Server:

watt.security.ssl.client.ignoreEmptyAuthoritiesList=true

After this watt property is set, CloudStreams will always send the trusted certificates irrespective of whether or not the server has sent the list of trusted authorities.

- **While enabling a connection, I get the following error: "javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target".**

  If connection enablement fails with "javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException" error, it implies that the certificate presented by the server during SSL Handshake has failed validation. This occurs when the server is using a self-signed certificate, or a certificate which is signed by a Certification Authority (CA) is not available in the truststore.

  Unless you have configured your own truststore using Integration Server Administrator under **Security > Certificates**, the certificates for most well-known commercial certificate authorities (CAs), for example, Verisign, GoDaddy, and so on, are available under the default or jvm truststore at: *<SAG_INSTALL>\jvm\jvm\jre\lib\security\cacerts* where *SAG_INSTALL* represents the Software AG installation directory. If the server is using a self-signed certificate or a CA certificate not available in the jvm truststore, import the self-signed or CA certificate into the jvm truststore. You can use the JSSE keytool for importing the self-signed or CA certificate.

Alternatively, if you have configured your own truststore using Integration Server Administrator under **Security > Certificates**, ensure that the CA certificates available in the jvm truststore are installed and available in the truststore configured by you.

- **Why do I get time out errors while enabling the connections or while executing connector services?**

  If connection enablement or service execution fails with time out related errors, for example, *Read Timeout* or *Timeout waiting for a connection*, configure the CloudStreams connections for time out values. In case the network is slow or the back end processing takes longer than usual, increase the *Connection Timeout* and the *Socket Read Timeout* values to 3 or 4 minutes. Based on the server responsiveness and network conditions, you may have to further increase or decrease this value. If you specify 0, the connection waits indefinitely but a value of 0 should be used only for debugging purposes and not in production environments because it can indefinitely block a connection. The correct time out values are not fixed and may vary based on the SaaS provider, load, network responsiveness, latency, and various other factors.

- **Why do I get "Time out or IOException" while trying to enable a connection?**

  A “Time out or IO Exception” maybe observed when you are behind a proxy server. Set up the proxy setting in Integration Server (**Settings > Proxy Servers**), and specify the *Proxy Alias* in the respective Connector configuration page. Select **Advanced View** under **Connection Groups: Connection** and specify the proxy alias against the **Proxy Server Alias** field. If a proxy is specified as the default proxy, then it will be automatically used. If you still see this issue, see the information provided in the “UNABLE_TO_RETRIEVE_CONNECTION_FROM_POOL: Unable to retrieve connection 'XXXX'. Cause: “Read timed out” and “Time out errors with retry” sections.

- **UNABLE_TO_RETRIEVE_CONNECTION_FROM_POOL: Unable to retrieve connection 'XXXX'. Cause: “Read timed out”**

  During execution of a connector service, you may sometimes observe the “Read timed out” exception.
  In a few cases the “Read timed out” exception can be a genuine exception when the response payload is large and reading the response payload might actually take more time than usual. To confirm the root cause, configure the CloudStreams Server loggers with the following suggested levels:
  - *0050 Runtime Trace*
  - *0300 Connection Factory Trace*
  - *0502 Connection Trace*
  - *0503 Service Trace*

  Observe the logs around the “Read timed out” error.
  If the “Read timed out” error is preceded by a partial payload log, increase the value of the Socket Read Timeout configuration.

However, if there is no response payload observed before the “Read timed out” error, and increasing the Socket Read Timeout to a sufficiently large value (say, 300,000 ms or 5 mins) does not help, it is possible that a stale connection is in use.

Enabling “Use Stale Checking” sometimes does not clean up stale connections. In order to enforce stale connection cleanup, use either of the below configurations. The configuration to be used depends on the connector service execution pattern during the “Read timed out” errors.

- If the connector service execution is continuous, and in between you observe a ‘Read timed out’ error, configure the Keep Alive Interval. For example, if you observe the ‘Read timed out’ error after 5 mins of continuous execution, configure the Keep Alive Interval to 240,000 ms or 4 mins. With the Keep Alive Interval configured to 4 mins, CloudStreams Server will discard a connection after 4 mins of usage.
- If the ‘Read timed out’ is observed after an idle period, configure the Idle Timeout. For example, if you observe the ‘Read timed out’ error after 5 mins of idle execution, configure the Idle Timeout to 240,000 ms or 4 mins. With the Idle Timeout configured to 4 mins, CloudStreams Server will discard a connection after its stays in the idle state for 4 mins.

- **Time out errors with retry**

  If the invocation of a connector service which is repeatable, that is, which can be executed multiple times without any side effects on the server side, *fails*, CloudStreams, with the following configuration, will try to execute it again till the service execution succeeds, or till it reaches the maximum number of retries (Connection Retry Count). For example, if Connection Retry Count is set as 3 and a repeatable service execution fails, CloudStreams will try to execute it for a maximum of 3 times or till it executes successfully.
  If you are facing an IO timeout issue and if the request is repeatable, enable the following connection configurations:

  1. Go to **Solutions > CloudStreams**.
  2. Click **Providers**.
  3. Select the connector. All connections under that connector are listed.
  4. Click **Yes** under the *Enabled* column to disable the connection.
  5. Click **Edit**.
  6. Click **Advanced view**.
  7. In the connection configuration properties page, **Connection** section, set **Retry on Response Failure** to **true**.
  8. Set **Connection Retry Count** to the number of times you want to retry in case of a failure. **Note**: This is true only for requests that are repeatable.

- **I configured “Retry on Response Failure”, but the errors do not resolve**

  CloudStreams has a retry mechanism in the *Advanced* connection configuration section. See the “Time out errors with retry” section for information. If the “Retry on Response Failure” option is set to true, it forces failed repeatable outbound requests to be retried.

Any repeatable requests should be retried till it succeeds or reaches the maximum configured number of “Connection Retry Count”, except in cases where the chunking is turned on. This means that for a connection, where “Use Chunking” is set to true, retrial of failed requests which are repeatable is not attempted.

If chunking is enabled, do the following to turn off chunking:

1. Go to **Solutions > CloudStreams**.
2. Click **Providers**.
3. Select the connector. All connections under that connector are listed.
4. Click **Yes** under the **Enabled** column to disable the connection.
5. Click **Edit**.
6. Click **Advanced** view.
7. In the connection configuration properties page, **Transport Protocol** section, set **Use Chunking** to false.
8. Save the changes and re-enable the connection.

   Note: If you want to send a large binary stream and if the back end supports HTTP chunking, it is recommended to set the “Use Chunking” option in the connection configuration page to true, but “Retry on Response Failure” will be irrelevant even if it is configured.

- **After I restart Integration Server, connector services that worked earlier, do not function any longer.**

  If you are using an OAuth connection with “fixed” Session Management mode, then after Integration Server is restarted, the connector service might not work.

  Check the authentication mechanism used by the connector. If it is OAuth, then check the back end error that you are receiving. If you get an “*Unauthorized: error with 401 error code*", that might indicate that the access token has already expired.
  To resolve this issue, set the Session Management option as “Auto”.

4 CloudStreams Server

- **Why does "java.lang.ClassCastException: com.softwareag.pg.pgmen.policy.config.LogPolicyInfo cannot be cast to com.softwareag.pg.pgmen.IPolicyInfo" exceptions appear in the server log and console?**

  If the custom package has no dependency on the provider package, the artifacts of the package will not be notified to the CloudStreams Server after the package reloads. The reason is that the different package classloader will load the package assets and with no dependency on the provider package, the assets will not be identified as meaningful CloudStreams nodes in Integration Server. Hence the connection nodes will not be visible. To fix this, from Software AG Designer, set the custom packages dependency on the provider package and then restart Integration Server.

- **I find only a partial Stack Trace in the Integration Server Administrator. How can I see the complete CloudStreams error Stack Trace?**

  To get the entire Stack Trace, enable the *watt.debug.layout* property and change the value from “legacy” to “new” using **Settings > Extended > Extended Settings** in Integration Server Administrator. You can also enable the **Connection Factory Wire Logging** option available under **Solutions > CloudStreams > Administration > General** tab. This option captures additional debug messages that show the request's HTTP content that is sent over the wire to the native provider. See the *Administering webMethods CloudStreams User Guide* for more information.

- **Integration Server response is slow.**

  If you have multiple unused connectors installed in Integration Server and observe degradation in Integration Server responsiveness, disable the document types packages for the unused connectors. This is recommended in case of large providers, for example, ERP providers like NetSuite™, which caters to a huge number of different types of requests. See the relevant CloudStreams connector specific user guide for more details on disabling document types packages for unused connectors.

- **Operations run for a longer duration and the SaaS provider returns an error.**

  In some cases, running **query** operations may take a long time to run on a SaaS provider like Salesforce. This maybe due to either the complexity of the query or the data involved in processing the query. In such cases, the SaaS provider displays an error.

  SaaS providers do not allow you to run queries indefinitely or for a very long time. In such cases, reduce the time taken for the operation to execute by optimizing the queries. See the relevant SaaS provider documentation on how to optimize the queries.

  You can also run those queries in any other alternate tool or user interface exposed by the SaaS provider.

- **Service requests are rejected for unsupported TLS versions.**

  For releases prior to CloudStreams version 9.12, if you want to use TLS Version 1.1 and 1.2 for CloudStreams connection requests, from the *Settings* menu in Integration Server Administrator, click *Extended* and add **watt.net.ssl.client.useJSSE=true** in the *Extended Settings* pane to use the latest TLS version. It is recommended to restart Integration Server. In most cases, only this property is required for SaaS providers like Salesforce.
  In some cases, where you want to force usage of any other protocol, for example, SSLv3, or a particular version of TLS properties, use the *watt.net.jsse.client.enabledProtocols* and *watt.net.ssl.client.handshake.minVersion* properties.

- **I am getting an error while invoking a connector service after upgrading from an older version of CloudStreams Server to a newer version.**

  Ensure that you run the **pub.cloudstreams.migration:migrate** service after upgrading to the newer version. The service will migrate your old CloudStreams artifacts to the latest version. For more information on the migration utility, see the Integration Server upgrade section in the *Upgrading Software AG Products* guide.

5 CloudStreams Governance

- **The CloudStreams event database gets filled up quite often than expected. How can I avoid this?**

  The default connector virtual service shipped by CloudStreams has log policies that log every outgoing request and incoming response. Due to this, the CloudStreams event database gets filled up quite often. To avoid this, define a custom virtual service and in the custom virtual service, do not add a log invocation policy. You can add a log invocation policy without the payload being logged to the database and server log.

- **CloudStreams event database is filled with gigabytes of data. My event database is full. How can I purge it and reduce my database size?**

  The default connector virtual service logs all outgoing requests and incoming responses by default. The request/response payloads could be large and it may accommodate a lot of space in the database. These request/response payloads are stored in the REQUEST/RESPONSE column of the CLS_TXN_EVENT table. To reduce the database size, in the CLS_TXN_EVENT table, purge the data in the REQUEST/RESPONSE column.

6 CloudStreams Services

- **Existing OData services, for example, C4C OData, fails and throws the following error at service runtime:**
  org.apache.olingo.odata2.core.edm.provider.EdmSimplePropertyImplProv cannot be cast to org.apache.olingo.odata2.api.edm.EdmNavigationProperty

  There is a possibility that your OData provider cloud instance has recently been patched or updated and because of this update, the OData services metadata may have changed. Contact your OData service provider and enquire about what has changed in the recent patch/update. The root cause of this error can be that the metadata received from the OData back end is corrupted.

7 CloudStreams Connector Listener

- **When I enable a connector listener in Integration Server Administrator, nothing seems to happen.**

  While enabling a CloudStreams connector listener, if there are connectivity issues like network and proxy issues while connecting to the streaming API endpoint, the connector listener automatically goes into retry mode and attempts to connect to the API endpoint until the configured connection timeout has been exhausted. The connector listener inherits the timeouts (Connection Timeout and Socket Read Timeout) from the referenced connector connection. In case the timeout is set to a large value, the update to the connector listener "enabled" status takes a long time to reflect in the Integration Server Administrator page. This may convey an impression that nothing is happening. To confirm the processing, check the server logs with the *Streaming* logging component configured to *Debug* or above. Alternatively, reduce the timeout values to speed up the "enabled" status update for the connector listener.

- **The status for my connector listener of connection type Bayeux HTTP Long Polling status shows as enabled, but my configured service invocation actions are not getting invoked after some time has elapsed since the listener was enabled.**

  Once a CloudStreams connector listener of connection type Bayeux HTTP Long Polling successfully transitions to an enabled state, and if at any point of time the connector listener receives an *advice={reconnect=none, interval=0}* from the back end, the listener disconnects and does not attempt to reconnect and re-establish the subscription. As a result, even though the connector listener remains in enabled state, the underlying subscription is no longer active.To confirm this as the root cause, scan the server logs for explicit */meta/disconnect meta-events*

  [CHANNEL:META_CONNECT]: {channel=/meta/disconnect}

  [CHANNEL:META_CONNECT]: {clientId=xiiusbhwbrxb2rqugi9xtq2xw, advice={reconnect=none, interval=0}, channel=/meta/connect, id=1028, error=400::Authenticated user id does not match the session's user

  Manually re-enable the connector listener to re-establish the subscription.

- **I sometimes see the following warnings in the server logs when connector listeners are operating in a clustered mode.**

  [CLS.0203.0100W] Thread interrupted when trying to obtain READ lock for the cache ListenerEventsCache

  net.sf.ehcache.constructs.nonstop.NonStopCacheException: get timed out

  at org.terracotta.modules.ehcache.store.nonstop.ExceptionOnTimeoutStore.get(ExceptionOnTimeoutStore.java:66)

  at org.terracotta.modules.ehcache.store.nonstop.NonStopStoreWrapper.get(NonStopStoreWrapper.java:800)

  at net.sf.ehcache.Cache.get(Cache.java:1749)

[CLS.0203.0100W] Thread interrupted when trying to obtain WRITE lock for the cache ListenerEventsCache

net.sf.ehcache.constructs.nonstop.NonStopCacheException: org.terracotta.toolkit.nonstop.NonStopException: createLockForKey timed out

at org.terracotta.modules.ehcache.concurrency.NonStopCacheLockProvider.getSyncForKey(NonStopCacheLockProvider.java:38)

at org.terracotta.modules.ehcache.ClusteredCacheInternalContext.getSyncForKey(ClusteredCacheInternalContext.java:22)

at net.sf.ehcache.Cache.getLockForKey(Cache.java:3950)

When a connector listener is active on multiple cluster nodes, the warnings could either be because of the following reasons:

- Terracotta server being down or unavailable, which results in all cache related operations (read and write) to time out. In this case, listeners on all the cluster nodes would continue to receive and process events agnostic of each other as in a standalone or non-clustered installation.
- When more than one connector listener instance proceeds to write to the Terracotta cache, there is a lock contention for the cache and only one instance can acquire the lock at a given time. Only when this instance completes the cache write operation, the other node will be able to acquire the lock to update the cache. Sometimes the connector listener instance times out waiting for the lock, which is when the above warnings appear.

- **After restarting Integration Server, listeners are not getting enabled. The listeners were all fine before the restart.**

  It can be due to the unavailability of the dependent connection at the time of the listener enablement. CloudStreams connector listeners are dependent on the connections they use to create and receive subscriptions. So before enabling the connector listeners, the connections must have been already registered and enabled. If both the connection and the listener artifacts reside in the same Integration Server package, then the order of loading the connection and then the listener is guaranteed. But this ordering cannot be assured if the connection and the listener are in different Integration Server packages, as the order of loading of the Integration Server packages is not guaranteed.

  So in a scenario where the connection and the listener artifacts stay across different Integration Server packages, and where there is no dependency set between them, it might so happen that the package containing the listener gets loaded first, does not find the dependent connection, and thus fails to get enabled. You must ensure package loading ordering by explicitly creating and declaring a package dependency, such that the package that contains the connector listener must depend on the package that contains the connection.

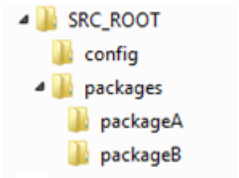
8 Building CloudStreams assets using webMethods Asset Build Environment

- **When I build CloudStreams assets using Asset Build Environment (ABE), I observe the following error regarding CloudStreams connections:**

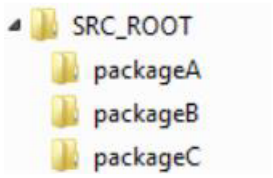
  [19/02/19 13:49][ERROR]: Passman store not initialized or unknown handle: com.softwareag.cloudstream.smcsalesforce.conn:SFConnection.cr.password

  The above error is an indication that ABE is unable to extract the value for the encrypted field, for example, the password field from the passman store using the passman handle specified in the error message. The root cause could be either a project directory structure inconsistent with the chosen build option because of which ABE is unable to resolve the Integration Server config directory containing the passman store or a missing config directory. Check if your project directory structure is consistent with the option specified during the ABE build.

  When you configure the **build.source.dir** property, packages containing the CloudStreams assets must be arranged as shown below, where SRC_ROOT is the configured path for the build.source.dir property. The config directory must be copied from the source Integration Server instances directory and placed parallel to the **packages** directory, directly under the SRC_ROOT directory.



  When you configure asset packages explicitly as a value to the **build.source.project.dir** property, your project directory structure will typically be organized as shown below. You must also specify a value for the **is.acdl.config.dir** property as the full path of the source Integration Server config directory.



9 Pre-requisites for Diagnostics

The following settings are recommended before collecting diagnostics for CloudStreams related issues.

1. Enable the *watt.debug.layout* property and change the value from “legacy” to “new” under **Settings > Extended > Extended Settings** in Integration Server Administrator. This will ensure that the entire stack trace is captured.

2. For issues specific to outbound SOAP/REST operation executions:

   - Configure the Logging Level under **Settings > Logging > View Server Logger Details** for the following *CloudStreams Facilities* to *Debug*.

     0300 Connection Factory
     0501 Connector
     0502 Connection
     0503 Service

   - Enable the *Connection Factory Wire Logging* option available under **Solutions > CloudStreams > Administration > General** tab. This option captures additional debug messages that show the request's HTTP content that is sent over the wire to the native provider.

   - Enable HTTP transport logging by adding the following entries in the *log_config.xml* file available at:
     For Windows:
     <Integration Server-Installation_dir>\profiles\IS_default\configuration\logging
     For Unix:
     <Integration Server-Installation_dir>/profiles/IS_default/configuration/logging

     <logger name="org.apache.http">
     <level value="debug" />
     </logger>

     <logger name="com.softwareag">
     <level value="debug" />
     </logger>

3. For issues specific to streaming exchange (COMET/HTTP Streaming):

   - Configure the *Logging Level* under **Settings > Logging > View Server Logger Details** for the following *CloudStreams Facilities* to *Debug*:

     0103 Streaming
     0504 Listener

   - Enable HTTP transport logging for streaming exchange by editing the following entry in the *log_config.xml* file available at:
     For Windows:
     <Integration Server-Installation_dir>\profiles\IS_default\configuration\logging
     For Unix:
     <Integration Server-Installation_dir>/profiles/IS_default/configuration/logging

```
<logger name="com.softwareag">
<level value="debug" />
</logger>
```

10 Other Considerations

This section covers other points that you must consider while working with CloudStreams.

- If you send multiple requests to the back end, CloudStreams will scale up to any level of concurrent usage provided you have enabled connection pooling and configured the connection pool in an optimal manner. This is true if there is no proxy involved.
  If a proxy intercepts the requests, ensure that you correctly configure the proxy to prevent it from acting as a bottleneck in the concurrent scale up.

- With newer authentication/authorization mechanisms evolving, some of the authentication mechanisms get deprecated. NT LAN Manager (NTLM) is one such mechanism and is not recommended to be used by the vendor.
  Hence, with CloudStreams, it is recommended to use a newer mechanism like OAuth, where ever it is available. Some of these older authentication mechanisms such as NTLM are not suitable for cloud based SaaS providers. They are also slower due to their handshake mechanisms and have more overhead in terms of performance. So it is recommended to use frameworks or mechanisms like OAuth, or the latest one recommended by the SaaS provider and supported by CloudStreams.

ABOUT SOFTWARE AG

Software AG offers the world's first Digital Business Platform. Recognized as a leader by the industry's top analyst firms, Software AG helps you combine existing systems on premises and in the cloud into a single platform to optimize your business and delight your customers. With Software AG, you can rapidly build and deploy Digital Business Applications to exploit real-time market opportunities. Get maximum value from big data, make better decisions with streaming analytics, achieve more with the Internet of Things, and respond faster to shifting regulations and threats with intelligent governance, risk and compliance. The world's top brands trust Software AG to help them rapidly innovate, differentiate and win in the digital world. Learn more at www.SoftwareAG.com.

software AG