



WEBMETHODS CLOUDSTREAMS PROVIDER FOR ALFABET INSTALLATION AND USER'S GUIDE

Version 9.10 | April 2016

CONTENTS

1	Document Change History	4
2	About this Guide	5
3	What is webMethods CloudStreams Provider for Alfabet?	6
3.1	What is a CloudStreams Connector?	6
3.2	What Is a Cloud Connection?	7
3.2.1	<i>Connection Pools</i>	7
3.2.2	<i>Run-Time Behavior of Connection Pools</i>	7
3.3	What Is a Cloud Connector Service?	8
4	Installing and Uninstalling CloudStreams Provider for Alfabet	9
4.1	Obtaining and Installing AlfaRest.jar	9
5	Managing CloudStreams Connectors	10
5.1	Enabling CloudStreams Connectors	10
5.2	Disabling CloudStreams Connectors	10
5.3	Deleting CloudStreams Connectors	10
5.4	Configuring the Properties of CloudStreams Connectors	11
6	Managing Cloud Connections	12
6.1	Creating Cloud Connections	12
6.2	Enabling Cloud Connections	22
6.3	Viewing Cloud Connections	22
6.3.1	<i>Sorting and Filtering Connections</i>	23
6.4	Disabling Cloud Connections	23
6.5	Editing Cloud Connections	24
6.6	Dynamically Changing a Cloud Service's Connection at Run Time	24
6.7	Copying Cloud Connections	24
6.8	Deleting Cloud Connections	25
7	Managing Cloud Connector Services	26
7.1	Creating a Cloud Connector Service	26
7.2	Editing a Cloud Connector Service for a REST-Based Provider	27
7.2.1	<i>Viewing the Constraints Applied to Variables</i>	31
8	CloudStreams Alfabet Connector	32
8.1	Overview	32
8.2	Connector Details	32
8.3	Manage Alfabet Connections	32

8.4	REST Resources	33
8.4.1	<i>Request and Response Processing</i>	33
8.4.2	<i>Input and Output Signature</i>	33
8.4.3	<i>Error and Fault Handling</i>	33
8.4.4	<i>Date Handling</i>	33
8.4.5	<i>Usage Notes</i>	34
8.4.6	<i>Supported Resources</i>	34
9	FAQ and Troubleshooting	37

1 Document Change History

Document Revision Date	Summary of Changes
April 2016	First release of this document.

2 About this Guide

This guide describes how to configure and use webMethods CloudStreams Provider for Alfabet. It contains information for administrators and application developers who want to interact with Alfabet to manage objects in Alfabet.

To use this guide effectively, you should be familiar with:

- Alfabet objects that you want to manage
- Terminology and basic operations of your operating system
- Setup and operation of the webMethods Integration Server
- Basic concepts and tasks of Software AG Designer

3 What is webMethods CloudStreams Provider for Alfabet?

webMethods CloudStreams Provider for Alfabet contains a predefined CloudStreams connector that you use to connect to an Alfabet instance. Using webMethods CloudStreams, you configure the CloudStreams Alfabet connector to manage security for inbound requests and log payloads, and to specify run-time performance conditions for consumers for outbound requests.

For more information about how to configure and use the CloudStreams connectors with webMethods CloudStreams, see the *Administering webMethods CloudStreams* document available in the webMethods section of the [Software AG Documentation](#) web page.

3.1 What is a CloudStreams Connector?

A CloudStreams connector contains connections and services that you use to integrate with a software as a service (SaaS) provider. The CloudStreams connector contains SaaS provider specific information, such as how to connect to a provider and the default and custom groups that you can configure. A CloudStreams connector is installed on Integration Server as an Integration Server package. When you start Integration Server for the first time, Integration Server dynamically creates the Integration Server document types for each of the out-of-the-box CloudStreams connectors installed with CloudStreams.

Each CloudStreams connector contains a *CloudStreams Connector Bundle* with all the assets required to enable the CloudStreams runtime to connect with the provider's back-end and to perform operations on the back-end. One of these assets is the *Cloud Connector Descriptor*, which contains:

- A reference to a metadata handler (SOAP or REST) that creates the data model representing the connector for a particular provider.
- Meta information that enables you to create the cloud connector's connections.
- The connector's SOAP operations or REST resources.

The type of a CloudStreams connector depends on the type of SaaS provider you communicate with. The two types of connectors are as follows:

- If you communicate with a SOAP-based cloud application provider, you create cloud connections using a CloudStreams SOAP connector.
- If you communicate with a REST-based provider, you create cloud connections using a CloudStreams REST connector.

You can use Integration Server Administrator to load, manage, and use the CloudStreams connectors. For a list of tasks that you must do before you can use a CloudStreams connector, see [Managing CloudStreams Connectors](#).

3.2 What Is a Cloud Connection?

You can create one or more connections for a CloudStreams connector at design time to use in integrations. A cloud connector service uses a cloud connection to connect to a SaaS provider at run time. You must create and enable a cloud connection before you can create cloud connector services. The number of connections you create depends on your integration needs.

You can configure connections using Integration Server Administrator. For a list of tasks you must do before creating connections, see [Managing Cloud Connections](#). For more information about configuring cloud connections, see [Creating Cloud Connections](#).

3.2.1 Connection Pools

Integration Server includes a connection management service that dynamically manages connections and connection pools based on configuration settings that you specify for the connection. By default, connection pooling is enabled for all cloud connections that you create.

A connection pool is a collection of connections with the same set of attributes. Integration Server maintains connection pools in memory. Connection pools improve performance by enabling cloud connector services to reuse open connections instead of opening new connections for every service request. All cloud connector services use connection pooling.

3.2.2 Run-Time Behavior of Connection Pools

When you enable a connection, Integration Server initializes the connection pool, creating the number of connection instances you specified in the connection's *Minimum Pool Size* field when you configured the connection. Whenever a cloud connector service needs a connection, Integration Server provides a connection from the pool. If no connections are available in the pool, and the maximum pool size has not been reached, the server creates one or more new connections (according to the number specified in the *Pool Increment Size* field) and adds them to the connection pool. If the pool is full (as specified in the *Maximum Pool Size* field), the requesting service will wait for Integration Server to obtain a connection, up to the length of time specified in the *Block Timeout* field, until a connection becomes available. Periodically, Integration Server inspects the pool and removes inactive connections that have exceeded the expiration period that you specified in the *Expire Timeout* field.

If the connection pool initialization fails because of a network connection failure or some other type of exception, you can enable the system to retry the initialization any number of times, at specified intervals. For information about configuring connection pooling for connections, see [Creating Cloud Connections](#).

3.3 What Is a Cloud Connector Service?

A cloud connector service is an Integration Server service used to integrate an on-premise application with a SaaS application. At run time, the cloud connector service constructs and maps a SOAP or REST request from the service's input pipeline to an appropriate message builder. When the cloud application provider sends a response, the cloud connector service processes and maps the response and populates the output pipeline.

You may need to create more than one cloud connector service to integrate an on-premise application with a cloud application. The number of cloud connector services you need to create depends on the type of provider you are communicating with and the description in that provider's Cloud Connector Descriptor:

- If you are using a SOAP-based cloud application provider, you create at least one cloud connector service for each operation defined in the Cloud Connector Descriptor. The operations contain a reference to a SOAP operation, defined in the connector's WSDL. For example, for a "query" operation, you might create a cloud connector service to query accounts and another to query contacts.
- If you are using a REST-based provider, you create at least one cloud connector service for each REST resource.

Creating a cloud connector service consists of two high-level steps:

1. Create a cloud connector service using a wizard in Software AG Designer. In this step, you give the service a name, select a CloudStreams Connector associated with a cloud provider, specify the cloud connection pool alias, and select the cloud virtual service that you want the cloud connector service to invoke. For details, see [Creating a Cloud Connector Service](#).
2. Edit the cloud connector service using the service editor in Software AG Designer. In this step, you specify the operation or REST resource of the service, the headers to include in the service, the input/output signature that determines how the user interacts with the service, and optional parameters to include in the input/output signature. For details about this step, see [Editing a Cloud Connector Service for a REST-Based Provider](#).

4 Installing and Uninstalling CloudStreams Provider for Alfabet

For installation instructions, see the *webMethods CloudStreams Provider Installation Guide* available on the [Software AG Documentation](#) website.

4.1 Obtaining and Installing AlfaRest.jar

CloudStreams Provider for Alfabet optionally supports using the "API" access mode to connect with the Alfabet REST API. To use this non-default mode, AlfaRest.jar is required. To use it, do the following:

1. Request and obtain the *AlfaRest.jar* file from the Alfabet Technical Consulting team at alfabet-techconsulting@softwareag.com.
2. Install the CloudStreams Provider for Alfabet.
3. Copy the *AlfaRest.jar* in the `/code/jars` directory:

```
<IntegrationServerDirectory>\instances\<instance_name>\packages\WmAlfabetProvider\code\jars
```

Example:

```
C:\SoftwareAG\IntegrationServer\instances\default\packages\WmAlfabetProvider\code\jars
```

4. Reload the WmAlfabetProvider package for the changes to take effect.
5. See [Creating Cloud Connections](#) (Connection Groups: Custom section) for instructions on enabling the API access mode.

5 Managing CloudStreams Connectors

You can manage the CloudStreams connectors using Integration Server Administrator. You must do the following tasks to prepare to configure connectors.

To prepare to configure a connector

1. Install webMethods Integration Server, webMethods CloudStreams server, and webMethods CloudStreams Provider for Alfabet on the same machine. For details, see the *Installing Software AG Products* guide and [Installing and Uninstalling CloudStreams Provider for Alfabet](#).
2. Ensure that you have webMethods administrator privileges so that you can access the CloudStreams connector administrative screens. For information about setting user privileges, see *webMethods Integration Server Administrator's Guide*.
3. Start Integration Server and Integration Server Administrator if they are not already running.
4. Using Integration Server Administrator, ensure that the cloud connector packages are enabled. See *webMethods Integration Server Administrator's Guide* for instructions.

5.1 Enabling CloudStreams Connectors

You must enable a CloudStreams connector before you can create and manage cloud connections to integrate with the connector's SaaS provider. You enable CloudStreams connectors using Integration Server Administrator.

To enable a cloud connector

1. In Integration Server Administrator, go to **Solutions > CloudStreams > Providers**.
2. Click the name of your cloud application provider, for example, **Software AG**.
3. On the **Connectors** screen, click **No** in the Enabled column for the connector you want to enable. Integration Server Administrator enables the connector and displays **Yes** in the Enabled column.

5.2 Disabling CloudStreams Connectors

You can disable CloudStreams connectors using Integration Server Administrator. You need to disable a connector before you can configure its properties or delete it.

To disable a cloud connector

1. In Integration Server Administrator, go to **Solutions > CloudStreams > Providers**.
2. Click the name of your cloud application provider, for example, **Software AG**.
3. On the **Connectors** screen, click **Yes** in the **Enabled** column for the connector you want to disable.

Integration Server Administrator disables the connector and displays **No** in the **Enabled** column.

5.3 Deleting CloudStreams Connectors

You can delete CloudStreams connectors using Integration Server Administrator.

Important! If you delete a connector, you can restore it only by reinstalling the connector.

To delete a cloud connector

1. In Integration Server Administrator, go to **Solutions > CloudStreams > Providers**.
2. Click the name of your cloud provider, for example, **Software AG**.
3. On the **Connectors** screen, ensure that the connector is disabled. To disable the connector, click **Yes** in the **Enabled** column.
4. Click the icon in the **Delete** column for the connector you want to delete.

Integration Server Administrator deletes the connector.

5.4 Configuring the Properties of CloudStreams Connectors

You can configure optional properties of a connector.

To configure the properties of a connector

1. In Integration Server Administrator, go to **Solutions > CloudStreams > Providers**.
2. Click the name of the cloud application provider you require, for example, **Software AG**.
3. On the **Connectors** screen, disable the connector you want to configure by clicking **Yes** in the **Enabled** column for that connector.
4. On the **Connectors** screen, click the icon in the **Configure** column.
5. On the **Connector Configuration** screen that appears, set the available properties for the connector as desired and click **Save**.

The properties are as follows:

Property	Description
Show connector document packages	<p>Hides/shows the document type packages that CloudStreams generated from the provider's WSDL or XSD.</p> <ul style="list-style-type: none"> • False (default): Hides the document type packages in the Software AG Designer user interface. • True: Shows the document type packages in the Software AG Designer user interface. You might want to show the document type packages during the development/testing phase for debugging purposes.

6. To view other details about the connector, click the icon in the **View** column on the **Connectors** screen.

The property settings take effect immediately.

6 Managing Cloud Connections

You can create and manage the cloud connections for each CloudStreams connector using Integration Server Administrator. You must do the following tasks to prepare to configure connections.

To prepare to configure a connection

1. Install webMethods Integration Server, webMethods CloudStreams, and the CloudStreams connectors on the same machine. For details, see the *Installing Software AG Products* guide and [Installing and Uninstalling CloudStreams Provider for Alfabet](#).
2. Ensure that you have webMethods administrator privileges so that you can access the webMethods CloudStreams connector administrative screens. For information about setting user privileges, see *webMethods Integration Server Administrator's Guide*.
3. Start Integration Server and Integration Server Administrator if they are not already running.
4. Using Integration Server Administrator, ensure that the CloudStreams connector packages are enabled. See *webMethods Integration Server Administrator's Guide* for instructions.
5. Using Software AG Designer, create a user-defined package to contain connections, if you have not already done so. See *webMethods Service Development Help* for instructions.

6.1 Creating Cloud Connections

You can create cloud connections for the installed and enabled CloudStreams connectors using Integration Server Administrator. For information about how to prepare for creating cloud connections, see [Managing Cloud Connections](#).

To create a connection

1. In Integration Server Administrator, go to **Solutions > CloudStreams > Providers**.
2. Click the name of the cloud application provider you require, for example, **Software AG**.
3. In the **Connector Name** column on the **Connectors** screen, click the name of the CloudStreams connector for which you want to create a connection.
4. On the **Connections** screen, click **Configure New Connection**.
5. On the **Configure Connection** screen, select in which view you want to create the connection:
 - **Basic view:** This is the default view. Use this view to configure the standard parameters for a cloud connection.
 - **Advanced view:** Use this view to configure additional and optional parameters for a cloud connection.

6. Complete the following standard parameters:

Section	Field	Description
	Package	<p>The package in which to create the connection. You must create the package using Software AG Designer before you can specify it using this parameter. For general information about creating and managing packages, see the <i>Designer Service Development online help</i>.</p> <p>By default, the connection is created in the Integration Server Default package.</p> <hr/> <p>Note: It is recommended that you configure the connection in a user-defined package. The custom package that you create must have a dependency on the WmCloudStreams package.</p> <hr/>
	Folder Name	The folder in which to create the connection. When the folder does not already exist in the package, Integration Server creates the folder automatically.
	Connection Name	The name of the new connection. Connection names cannot have spaces or use special characters reserved by Integration Server or Software AG Designer. For more information about the use of special characters in package, folder, and element names, see the <i>Designer Service Development online help</i> .
Connection Groups: Connection	Server URL	<p>The login endpoint to initiate communication with the SaaS provider.</p> <p>The URL for Alfabet REST connector depends on where the required instance of alfabet is installed. It is possible to either include or omit the endpoint suffix <code>"/Alfabet/api/vXX"</code> in the URL.</p> <p>For example, both these options are equivalent:</p> <ul style="list-style-type: none"> • <code>https://myalfabet.com</code> • <code>https://myalfabet.com/Alfabet/api/v1</code>
Connection Groups: Transport Protocol	Element Character Set	<p>The encoding to use for the HTTP message components, such as request line and headers.</p> <p>Default: US-ASCII</p>

Section	Field	Description
	Strict Transfer Encoding	Whether the connection validates the HTTP Transfer Encoding header. Valid values: true : The connection validates the Transfer Encoding header and returns an error when the header is invalid. false : The connection does not validate the Transfer Encoding header. Default: false
Connection Groups: Custom	Authorization Token	The Alfabet Authorization Token as defined in the web.config file of the Alfabet Web Application on the server side, under the <a1faSection> element.
	Access Mode	The connector uses the Alfabet HttpClient access mode by default. To use the API access mode, AlfaRest.jar is required. See Obtaining and Installing AlfaRest.jar before using the API.
Connection Management Properties	Enable Connection Pooling	Whether connection pooling is enabled for a connection. Valid values: <ul style="list-style-type: none"> true: Connection pooling is enabled for this connection. false: Connection pooling is disabled for this connection. Default: true
	Initial Pool Size	The minimum number of connection objects that remain in the connection pool at all times, if connection pooling is enabled. When the connector creates the pool, it creates this number of connections. Default: 1
	Maximum Pool Size	The maximum number of connection objects that can exist in the connection pool if connection pooling is enabled. When the connection pool has reached its maximum number of connections, the connector will reuse any inactive connections in the pool, or, if all connections are active, it will wait for a connection to become available. Default: 10

Section	Field	Description
	Pool Increment Size	<p>The number of connections by which the pool will be incremented, up to the maximum pool size, if connection pooling is enabled and connections are needed.</p> <p>Default: 1</p>
	Block Timeout (msec)	<p>The number of milliseconds that Integration Server will wait to obtain a connection with the SaaS provider before the connection times out and returns an error.</p> <p>For example, you have a pool with Maximum Pool Size of 20. If you receive 30 simultaneous requests for a connection, 10 requests will be waiting for a connection from the pool. If you set the Block Timeout to 5000, the 10 requests will wait for a connection for 5 seconds before they time out and return an error. If the services using the connections require 10 seconds to complete and return connections to the pool, the pending requests will fail and return an error message stating that no connections are available.</p> <p>If you set the Block Timeout value too high, you may encounter problems during error conditions. If a request contains errors that delay the response, other requests will not be sent. This setting should be tuned in conjunction with the Maximum Pool Size to accommodate such bursts in processing.</p> <p>Default: 1000</p>
	Expire Timeout (msec)	<p>The number of milliseconds that an inactive connection can remain in the pool before it is closed and removed from the pool, if connection pooling is enabled.</p> <p>The connection pool will remove inactive connections until the number of connections in the pool is equal to the Initial Pool Size. The inactivity timer for a connection is reset when the connection is used by the connector.</p> <p>This setting should be tuned in conjunction with the Initial Pool Size to avoid excessive opening/closing of connections during normal processing.</p> <p>Default: 1000</p>

Section	Field	Description
	Startup Retry Count	<p>The number of times that the system should attempt to initialize the connection pool at startup if the initial attempt fails.</p> <hr/> <p>Note: The retry mechanism is invoked only when the connection is configured correctly, but the target server URL cannot be reached or a network issue occurs while attempting to initialize the connection.</p> <hr/> <p>Default: 0 (a single attempt)</p>
	Startup Backoff Timeout (sec)	<p>The number of seconds that the system should wait between attempts to initialize the connection pool. This value is ignored if Startup Retry Count is 0.</p> <p>Default: 10</p>

Section	Field	Description
	Session Management	<p>The type of timeout for the connection session. Select the type of session management that fits the requirements of your SaaS provider backend.</p> <p>In case of Alfabet, it is recommended that you set this field to none because each service call is authenticated separately, and there is not concept of session.</p> <p>Valid values:</p> <ul style="list-style-type: none"> none: The CloudStreams server does not manage session timeout. The session times out based on the settings of the SaaS provider backend. idle: If no activity happens for the time specified in Session Timeout, the session times out. If the session is not idle (it is used actively), the session will not timeout. The CloudStreams server takes into account the idle timeout. For example, if the session is idle for the time specified in Session Timeout, the server renews the session before making the service call. fixed: The session will timeout at a fixed time interval (specified in Session Timeout) irrespective of the session usage or current activity. The CloudStreams server renews the session as soon as the fixed timeout value expires.
	Session Timeout (min)	<p>Not applicable if session management is none.</p> <p>The maximum number of minutes a session can remain active (in other words, how long you want the server to wait before terminating a session). The value should be equal to the session timeout value specified at the SaaS provider backend.</p>

7. If you selected **Advanced view**, complete the following fields in addition to the fields you configured in the **Basic view**:

Important! If you do not want to use **Advanced view**, skip this step and continue with *Step 8*.

Section	Field	Description
---------	-------	-------------

Section	Field	Description
Connection Groups: Connection	Connection Timeout	<p>The number of milliseconds a connection waits before canceling its attempt to connect to the resource. If you specify 0, the connection waits indefinitely.</p> <hr/> <p>Important! It is recommended that you specify a value other than 0 to avoid using a socket with no timeout.</p> <hr/> <p>Default: 30000</p>
	Socket Read Timeout	<p>The number of milliseconds in which the client must read a response message from the server. If you specify 0, the connection waits indefinitely.</p> <hr/> <p>Important! It is recommended that you specify a value other than 0 to avoid using a socket with no timeout.</p> <hr/> <p>Default: 30000</p>
	Use Stale Checking	<p>Whether the connection performs additional processing to test if the socket is still functional each time the socket is used.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • true: The connection tests the socket. • false: The connection does not test the socket. <hr/> <p>Note: The additional testing of the socket adds a little performance overhead.</p> <hr/> <p>Default: false</p>
	Connection Retry Count	<p>The number of times the system should attempt to initialize the connection at startup if the initial attempt fails.</p> <p>The system retries to establish a connection when an I/O error occurs while sending the request message to the backend. If an I/O exception occurs when the system is reading a response back from the backend, the system will only retry if Retry on Response Failure is set to true.</p> <p>Default: 1</p>

Section	Field	Description
	Retry on Response Failure	<p>Whether the system should attempt to resend the request when the response has failed, even though the request was sent successfully.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • true: The system attempts to reestablish the connection. • false: The system does not attempt to re-establish the connection. <p>Default: false</p>
	Use TCP NoDelay	<p>Whether to use algorithm to reduce the number of packets that need to be sent over the network.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • true: Do not optimize the bandwidth consumption. • false: Use Nagle's algorithm to optimize the socket usage. <p>Default: false</p>
	Socket Linger	<p>The number of seconds before a client socket closes.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • -1: Use the Java VM default. • 0: Close the socket connection immediately. • $n > 0$: Wait for n seconds before closing the socket connection. <p>Default: -1</p>
	Socket Buffer Size	<p>The size of the read and write socket buffers in bytes.</p> <p>Default: 8192</p>
	Socket Reuse Address	<p>Whether the socket will be reused even if it is in the TIME_WAIT state because of a previous socket closure.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • true: The socket will be reused. • false: The socket will not be reused. <p>Default: false</p>

Section	Field	Description
	Proxy Server Alias	<p>The alias name of an enabled proxy server configuration on Integration Server that will be used to route the connection.</p> <hr/> <p>Note: When the corresponding proxy server configuration on Integration Server is updated, the changes are detected automatically. You do not need to re-enable the connection to use the updated proxy server configuration.</p> <hr/>
	Trust Store Alias	<p>The alias name of an Integration Server trust store configuration. The trust store contains trusted certificates used to determine trust for the remote server peer certificates.</p> <hr/> <p>Note: Setting the Integration Server <code>watt.security.cert.wmChainVerifier.trustByDefault</code> property to "true" overrides the value in this field. In this case, the server will trust all remote server peer certificates. If you want to use the Trust Store Alias field, set the Integration Server <code>watt.security.cert.wmChainVerifier.trustByDefault</code> property to "false".</p> <hr/>
	Hostname Verifier	<p>The fully qualified classname of the Apache X509HostnameVerifier interface.</p> <p>Default: org.apache.http.conn.ssl.StrictHostnameVerifier</p> <p>When you configure strict hostname enforcement, the connection verifies whether the server certificate matches the server host. If you do not specify a value in this field, the connection uses the <code>org.apache.http.conn.ssl.AllowAllHostnameVerifier</code> that disables hostname enforcement.</p> <hr/>
Connection Groups: Protocol	HTTP Content Character Set	<p>The encoding to use for the HTTP request message.</p> <p>Default: ISO-8859-1</p> <hr/>

Section	Field	Description
	HTTP Protocol Version	<p>The version of the HTTP transport protocol that the connection will use.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • HTTP/0.9 • HTTP/1.0 • HTTP/1.1 <p>Default: HTTP/1.1</p>
	User Agent	<p>The name of the client that the connection includes in the HTTP User-Agent request header to identify the origin of the request.</p> <p>Default: CloudStreams</p>
	Use Expect Continue	<p>Whether to use the Expect/Continue HTTP/1.1 handshake and send the Expect request header. When the client sends the Expect request header, the client waits for the server to confirm that it will accept the request, before the client sends the request body.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • true: Use the Expect/Continue handshake. • false: Do not use the Expect/Continue handshake. <p>Default: false</p>
	Wait for Continue Time	<p>The number of milliseconds that the client connection should wait for a 100 Continue response from the server when the Expect/Continue handshake is used.</p> <p>Default: 3000</p>
	Use Chunking	<p>Whether to use HTTP/1.1 chunking with a chunk size that matches the socket buffer size.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • true: Use HTTP/1.1 chunking. • false: Do not use HTTP/1.1 chunking. <p>Default: false</p>

Section	Field	Description
	Follow Server Redirects	Whether the connection follows server redirects. Valid values: <ul style="list-style-type: none"> true: The connection follows server redirects. false: The connection does not follow server redirects. Default: true
	Server Redirect Maximum Tries	The number of times to allow a request to be redirected before the server returns an I/O exception to the client. Default: 5

8. Click **Save**.

You must enable a cloud connection before you can use it. For information about how to enable a connection, see [Enabling Cloud Connections](#).

6.2 Enabling Cloud Connections

You must enable a cloud connection before you can use the CloudStreams connector associated with the cloud connection to create cloud connector services and integrate with the cloud application provider. You enable cloud connections using Integration Server Administrator.

Note: When you reload a package that contains enabled connections, the connections will automatically be enabled when the package reloads. If the package contains connections that are disabled, they will remain disabled when the package reloads.

To enable a connection

1. In Integration Server Administrator, go to **Solutions > CloudStreams > Providers**.
2. Click the name of the cloud application provider you require, for example, **Software AG**.
3. On the **Connectors** screen, click the **Connector Name** for the CloudStreams connector whose connection you want to enable.
4. On the **Connections** screen, click **No** in the **Enabled** column for the connection you want to enable.

Integration Server Administrator enables the connection and displays **Yes** in the **Enabled** column.

6.3 Viewing Cloud Connections

You can view cloud connections and each connection's parameters from the Integration Server Administrator.

Note: You can also view this information from the cloud connection editor in the Service Development perspective in Software AG Designer.

To view a connection

1. In the Integration Server Administrator, go to **Solutions > CloudStreams > Providers**.
2. Click the name of the cloud application provider you require, for example, **Software AG**.
3. On the **Connectors** screen, click the **Connector Name** for the CloudStreams connector whose connection you want to view.
4. On the **Connections** screen, click the icon in the **View** column for the connection you want to view.
5. On the **View Connection** screen, select in which view you want to view the connection parameters:
 - **Basic view:** This is the default. Use it to view the standard parameters for a cloud connection.
 - **Advanced view:** To view additional parameters for a cloud connection, click the **Advanced view** link.

The **View Connection** screen displays the parameters for the connection. For descriptions of the connection parameters, see the table of parameters in [Creating Cloud Connections](#).

6.3.1 *Sorting and Filtering Connections*

You can sort and filter the list of connections that appears on the **Connections** screen.

To sort and filter connections

1. To sort information on the **Connections** screen, click the **Up** and **Down** arrows in each column.
2. To filter the list of connections:
 - a. On the **Connections** screen, click **Filter Connections**.
 - b. Type the criterion by which you want to filter, into the **Filter criteria** box. Filtering is based on the connection alias. To locate all connections containing specific alphanumeric characters, use asterisks (*) as wildcards. For example, if you want to display all connections containing the string "abc", type *abc* in the **Filter criteria** box.
 - c. Click **Search**. The **Connections** screen displays the connections that match the filter criteria.
 - d. To redisplay all connections, click **Show All Connections**.

The **Connections** screen appears, listing all the current connections.

6.4 **Disabling Cloud Connections**

You can disable cloud connections when you want to edit or delete them using the Integration Server Administrator.

To disable a connection

1. In the Integration Server Administrator, go to **Solutions > CloudStreams > Providers**.
2. Click the name of the cloud application provider you require, for example, **Software AG**.
3. On the **Connectors** screen, click the **Connector Name** for the CloudStreams connector whose connection you want to disable.

4. On the **Connections** screen, click **Yes** in the **Enabled** column for the connection you want to disable.
5. Integration Server Administrator disables the connection and displays a **No** in the **Enabled** column.

6.5 Editing Cloud Connections

If a connection parameter changes, or if you want to redefine parameters that a connection uses when connecting to a cloud application provider, you can update a connection's parameters using the Integration Server Administrator.

To edit a connection

1. In Integration Server Administrator, go to **Solutions > CloudStreams > Providers**.
2. Click the name of the cloud application provider you require, for example, **Software AG**.
3. On the **Connectors** screen, click the **Connector Name** for the CloudStreams connector whose connection you want to edit.
4. Ensure that the connection is disabled before editing. To disable the connection, click **Yes** in the **Enabled** column. The **Enabled** column now shows **No** (disabled) for the connection.
5. On the **Connections** screen, click the **Edit** icon for the connection you want to edit. The **Edit Connection** screen displays the current parameters for the connection. Select the view in which you want to edit and update the connection's parameters by typing or selecting the values you want to specify.

For descriptions of the connection parameters, see [Creating Cloud Connections](#).

6. Click **Save Changes**.

6.6 Dynamically Changing a Cloud Service's Connection at Run Time

You can dynamically select the cloud connection a cloud service uses, to interact with a cloud application. You can run a cloud service using a cloud connection other than the default connection that was associated with the cloud service when the service was created. This feature enables one cloud service to interact with multiple, similar cloud applications.

To override the default cloud connection, you must code your flow to pass a value through the pipeline into a service's *\$connectionAlias* field.

6.7 Copying Cloud Connections

You can copy an existing cloud connection to configure a new connection with the same or similar connection properties without having to retype all of the properties for the connection. You copy cloud connections using the Integration Server Administrator.

To copy a connection

1. In Integration Server Administrator, go to **Solutions > CloudStreams > Providers**.
2. Click the name of the cloud application provider you require, for example, **Software AG**.

3. On the **Connectors** screen, click the **Connector Name** for the CloudStreams connector whose connection you want to copy.
4. On the **Connections** screen, click the **Copy** icon for the connection you want to copy. The **Copy Connection** screen displays the current parameters for the connection you want to copy.
5. Name the new connection, specify a package name and folder name, and edit any connection parameters as needed by typing or selecting the values you want to specify.

Note: When you copy a connection, the new connection does not save the password of the original connection. You must enter the password before you can save the new connection.

For descriptions of the connection parameters, see [Creating Cloud Connections](#).

6. Click **Create**.

6.8 Deleting Cloud Connections

You can delete cloud connections using the Integration Server Administrator.

To delete a connection

1. In the Integration Server Administrator, go to **Solutions > CloudStreams > Providers**.
2. Click the name of the cloud application provider you require, for example, **Software AG**.
3. On the **Connectors** screen, click the **Connector Name** for the CloudStreams connector whose connection you want to delete.
4. Ensure that the connection is disabled before deleting. To disable the connection, click **Yes** in the **Enabled** column. The **Enabled** column now shows **No (disabled)** for the connection.
5. Click the **Delete** icon for the connection you want to delete.

Integration Server deletes the cloud connection.

7 Managing Cloud Connector Services

You can create and manage cloud connector services using Software AG Designer.

Keep the following points in mind when creating a cloud connector service:

- Before you create a cloud connector service, ensure that the CloudStreams connector associated with your desired cloud application provider is installed. Also ensure that a cloud connection pool is created for that connector.
- You should create at least one cloud connector service for each REST resource. For more information about REST resources, see [CloudStreams Alfabet Connector](#).

7.1 Creating a Cloud Connector Service

You can create a cloud connector service using Software AG Designer.

To create a cloud connector service

1. Open the Service Development perspective in Software AG Designer if it is not already open.
2. Navigate to and expand the package in which you want the cloud connector service to reside. Right-click the folder in which you want to create the service and select **New > Cloud Connector Service**.

Software AG Designer displays the **New Cloud Connector Service** wizard.

3. On the **Cloud Connector Service** page of the wizard, in the **Element name** field, type the name you want to assign to the cloud connector service and click **Next**.
4. On the **Connector** page of the wizard, select the CloudStreams Connector associated with the cloud application provider you want to access and click **Next**.

Tip! If the list of available connectors is long and you know the name of the connector you want to use, you can locate the connector quickly by typing its name in the box below **Available Connectors**. You can also use this technique when selecting the connection pool and service in the next steps.

5. On the **Connection Pool** page of the wizard, select the connection pool for connecting to the cloud application provider and click **Next**.
6. On the **Select Service** page of the wizard, select the cloud virtual service that you want the cloud connector service to invoke.

Note: If only one cloud virtual service is available to select, this page will not appear.

7. Click **Finish**.

Software AG Designer creates the cloud connector service and displays the service details in the cloud connector service editor.

8. Edit the cloud connector service as follows:

For this type of provider...	Follow the steps described in...
REST-based	Editing a Cloud Connector Service for a REST-Based Provider

7.2 Editing a Cloud Connector Service for a REST-Based Provider

Editing a cloud connector service for a REST-based provider consists of specifying the resource, the type of processing for requests or responses, the headers to include in the service, the input/output signature that determines how the user interacts with the service, default values for parameters included in the input/output signature, and descriptive comments or usage notes, if any. You can edit a cloud connector service using the service editor in Software AG Designer.

Keep the following points in mind when editing a cloud connector service:

- Before you edit a cloud connector service, create the service as described in [Creating a Cloud Connector Service](#).
- webMethods CloudStreams provides a default connector virtual service for policy enforcements, called *WmCloudStreams.RestVS*. If this service does not meet the needs of your CloudStreams project, ensure that an appropriate connector virtual service has been created for your project. For more information about CloudStreams connector virtual services, see the *Administering webMethods CloudStreams* document.
- In pipeline, document, and input/output validation, the data validation applies constraints to its variables. Constraints are the restrictions on the structure or content of variables. For more information about icons for constrained variables, see [Viewing the Constraints Applied to Variables](#).

To edit a cloud connector service for a REST-based provider

1. Open **Software AG Designer** if it is not already open.
2. Navigate to and open the cloud connector service you created in [Creating a Cloud Connector Service](#). The service opens in the cloud connector service editor.
3. On the **Resource** tab, do the following:
 - a. From the **Connector Virtual Service** list, select the connector virtual service to be used for policy enforcements. For more information about CloudStreams connector virtual services, see the *Administering webMethods CloudStreams* document.
 - b. Click  next to **Resource Name**. Software AG Designer displays the **Resource and Business Object Configuration** wizard.
 - c. Select the REST resource you want the cloud connector service to process, and then click **Next**. When you change a resource, Software AG Designer clears all the metadata that were associated with the previously selected resource, including the headers, parameters, and data types of fields. You can select the metadata that the updated resource requires in the next steps.

Note: Software AG Designer displays the appropriate pages of the Resource and Business Object Configuration wizard depending on whether the selected resource requires metadata, such as a business object, fields, and data types of fields.

- d. In the **Select the Business Object** page, select a business object and click Next.
- e. In the **Select Fields** page, specify the fields or parameters to use in the request/ response body for the object. The mandatory fields or parameters for the business object are selected by default, and cannot be cleared.
- f. If you want to configure concrete types for the abstract types in the resource you selected, click Next. If the resource you selected does not have any abstract type field, click Finish.
- g. In the **Configure Data Types of Fields** page, select a value from the list of values next to the abstract type to configure concrete types for the abstract types in the resource.
- h. Click Finish. Software AG Designer displays a confirmation message. Click OK to update the resource. Software AG Designer replaces the existing resource and associated metadata with the updated or default information.
- i. In the **Request Processing** section, select an appropriate parsing type. The parsing type determines how the service accepts the input.

Option	Meaning
Document	Builds the request message as an IS document type. Select this option when the provider's XML file includes a schema or specification describing the content of the request.
Binary Stream	Builds the request message as a binary stream. Select this option when you expect the pipeline to contain an input stream for which no document type exists or when it is not practical to provide a schema description of the content.

Note: If the resource you selected does not contain any requests or responses, the Request Processing or Response Processing fields are not available.

- j. In the **Response Processing** section, select an appropriate serialization type. The serialization type constructs the cloud connector service's output signature and determines how the cloud connector service should return data to the user.

Option	Meaning
Document	Formats the response message as an IS document type. Select this option when the provider's XML file includes a schema or specification describing the content of the response.
Binary Stream	<p>Formats the response message as a binary stream. Select this option when you expect the pipeline to contain an output stream for which no document type exists or when it is not practical to provide a schema description of the content.</p> <p>Note: This option works in conjunction with the response's parsing type property. If you select Stream as the response's serialization type, Software AG Designer also selects Stream as the response's parsing type.</p>

Note: If the resource you selected does not contain any requests or responses, the Request Processing or Response Processing fields will not be available.

4. On the **Headers** tab, Software AG Designer displays the default HTTP transport headers for the resource, along with their default values. At run time, while processing the headers, CloudStreams substitutes values, if necessary. In order to customize the headers, do the following:
- To specify a default value for the header variable, click the **Default Value** box to the right of the variable and type or paste the new value. If the variable is null in the input pipeline, this value will be used at run time. If the variable has an existing default value defined in the Cloud Connector Descriptor, this value will overwrite the existing value at run time. However, if the existing default value is of type *fixed default*, overwrite will fail as mentioned earlier.
 - To add a custom header to the service's input pipeline, in the **Input** section of the tab, click . Type a name for the header and provide a default value if desired.
 - To move a header up in the list, select the header and click . To move a header down in the list, select the header and click .
 - To include a header as part of the service signature, select the **Active** check box next to the header.
 - To delete a custom header that you added, select the header and click .

Note: You cannot delete the resource's required headers.

- f. Repeat the above steps in the **Output** section of the tab to select the HTTP transport protocol headers whose contents you want to add to the service's output pipeline.

Note: A provider's response headers appear in the pipeline signature only if they are added as active output headers in the **Output** section. Any unspecified headers returned by the native provider will not be included in the pipeline.

5. On the **Parameters** tab, Software AG Designer displays the configured resource parameters. In order to customize the parameters, do the following:
- Review the details about the resource parameters. Software AG Designer displays the parameter name and description, the data type used to represent the kind of information the parameter can hold, the parameterization style of the request, and the dynamic default value needed to access the resource.

Currently, three parameter styles are supported: *URI_CONTEXT*, *QUERYSTRING_PARAM*, and *CFG_PARAM*.

For more information about the supported parameter styles, see the section *Understanding REST Parameters* in the *Administering webMethods CloudStreams* document.

- To specify a default value for the parameter, click the **Default Value** box to the right of the parameter. Then, type or paste the default value. The default value is used at run time, if the parameter value is not explicitly specified in the input pipeline. Also, this default value will overwrite any existing default value that is defined in the Cloud Connector Descriptor, at run time. However, if the existing default value is of type *fixed default*, overwrite will fail as mentioned earlier.

Note: You cannot specify a default value for a parameter with data type as *Record*.

6. On the **Input/Output** tab, do the following:
- To have the server validate the input to the service against the service input signature, select the **Validate input** check box.
 - To have the server validate the output to the service against the service output signature, select the **Validate output** check box.
 - Review the service's input and output signature and make any necessary changes as follows:

To change the...	Go to the...
List of headers in the requestHeaders or responseHeaders section, or their default values	Headers tab
Default value of a parameter in the parameters section, or their default values	Parameters tab

The requestBody and responseBody sections are derived from the REST resource you selected on the Resource tab. The value of *\$connectionAlias* is derived from the connection pool you specified when you first created the cloud connector service. The status, statusMessage, and fault values are derived from the resource response. You cannot change these values in the editor.

7. On the **Logged Fields** tab, do the following:
- Select the check boxes next to the fields you want to log at run time.
 - If you want to create an alias for a logged field to make it easier to locate in Software AG Designer, click the **Alias** box next to a field and type the alias name.

For more information about logged fields, see the section on logging input and output fields in Software AG Designer.

8. On the **Summary** tab, review the details about the cloud connector service.

9. On the **Comments** tab, enter descriptive comments or usage notes, if any.
Click **File > Save** to save your changes.

7.2.1 Viewing the Constraints Applied to Variables

Software AG Designer displays small symbols next to a variable icon to indicate the constraints applied to the variable. Software AG Designer displays variables in the following ways:

Variable	Constraint Status	Variable Properties
 String	Required field.	The Required property is set to True.
 String	Optional field.	The Required property is set to False.
 String	Required field with content type constraint.	The Content type property specifies an IS schema or XML schema.
 String	Optional field with content type constraint.	The Required property is set to False, and the Content type property specifies an IS schema or XML schema.
 String	Required field with default value.	The Fixed property is set to False, and the defaultValue property specifies a default value. The variable has a default value, but you can override this default value with any other valid values while executing the service or mapping the variables.
 String	Required field with fixed value.	The Fixed property is set to True, and the defaultValue property specifies a null value. The variable has a null value assigned to it by default and you cannot override this value. You cannot map this variable to another variable or assign any input values to this variable during service execution.
 String	Required field with fixed default value.	The Fixed property is set to True, and the defaultValue property specifies a default value. The variable has a default value and you cannot override this value. You cannot map this variable to another variable or assign any input values to this variable during service execution.

8 CloudStreams Alfabet Connector

8.1 Overview

CloudStreams Alfabet connector uses REST resources for managing objects in the Alfabet database.

The following sections describe only the basic information you need, to use the REST resources supported by the CloudStreams Alfabet connector.

For detailed information about the underlying REST resources, see the *Alfabet Interface for RESTful Web Services* reference manual.

8.2 Connector Details

The connector details include:

- **SaaS Provider:** Software AG
- **Connector Name:** Alfabet
- **API Version:** 1
- **API Type:** REST
- **Developer:** Software AG
- **Group:** Software AG
- **CloudStreams Minimum Version Compatibility:** 9.10
- **Provider Package Name:** WmAlfabetProvider

8.3 Manage Alfabet Connections

Alfabet supports the following access modes:

- **HttpClient**
- **API**

By default, CloudStreams Alfabet provider uses **HttpClient** access mode after you have provided the connection with an Authorization Token. See [Creating Cloud Connections](#) for more details.

To use the **API** access mode, Alfabet requires a new token generated per request. The CloudStreams Alfabet provider takes care of that once you provide the connection with an Authorization Token, and the `AlfaRest.jar` from Alfabet. See [Obtaining and Installing AlfaRest.jar](#) for relevant instructions.

See the "Authorization" chapter in the *Alfabet Interface for RESTful Web Services* reference manual for required configurations in the server side for Alfabet, and for details about the different authorization modes. Being designed for integration purposes, CloudStreams Alfabet provider does not support the optional per user token.

8.4 REST Resources

8.4.1 Request and Response Processing

The Cloud connector contains the expected Request and Response default values for each of the REST resources. When you create a Cloud Connector Service in Software AG Designer, you either use the default values (recommended when you are not sure what values are required) or select Request and Response values from a drop-down list.

For CloudStreams connector, the request and response processing type can be set either to **Document** or **Binary Stream**. See [Editing a Cloud Connector Service for a REST-Based Provider](#) for more information on setting **Request Processing** and **Response Processing** types.

8.4.2 Input and Output Signature

The Alfabet connector contains an IS document that maps to the request or response data for a REST resource, for example, *wmAlfabetConnector_vXX.doctypes:createObjectRequest*, where *XX* is the version of the Alfabet connector.

The input values to the fields in the service signature map to the input values required by the underlying Alfabet REST API. For more information about the required fields to configure for a request, see the Alfabet REST API documentation.

Where base64 encoding is required, the Alfabet connector is encoding the requests.

8.4.3 Error and Fault Handling

The Alfabet REST API v1 error codes and messages return within the response documents and do not have separate error docTypes. The http response is always 200 (OK).

To handle errors, the first instance of the response array can be observed. Here are the relevant fields:

- **Kind**: "RESULT".
- **Status**: "ERROR" (or, "OK" in case of success).
- **Code**: Number (typically over 900) indicating the error code. See Error Codes of the Alfabet RESTful API in the *Alfabet Interface for RESTful Web Services* reference manual.
- **Message**: Explanation about the source of the error.

Note: *Code* and *Message* might not be populated in case of success.

8.4.4 Date Handling

Alfabet connector uses the following format for any date datatype: yyyy-MM-dd, for example, 2016-10-20.

8.4.5 Usage Notes

For information about how to access the description of a REST resource along with its details in Software AG Designer, see [Editing a Cloud Connector Service for a REST-Based Provider](#).

8.4.6 Supported Resources

The following sections describe the REST resources supported by the Alfabet connector. For detailed information about the underlying REST resources, see the *Alfabet Interface for RESTful Web Services* reference manual.

Retrieve

This resource retrieves the information stored about a single object in the Alfabet database based on its REFSTR. When defining the Cloud Connector Service, select the class of the object to retrieve. All the fields of the class will be automatically selected for retrieval.

The following parameters are available:

- *refstr*: required, the Alfabet REFSTR of the required object.
- *emptyvalues*: optional (default: true), indicated whether to retrieve properties that have no value.
- *language*: optional, language code, in case Alfabet instance translation is in use.

Resource Definitions

wmAlfabetConnector_vXX.doctypes:retrieveResponse

This resource invokes an HTTP GET method against the *object* endpoint of Alfabet.

Query

This resource returns information about objects of a specific Class that is the result of a native SQL query on the Alfabet database. When defining the Cloud Connector Service, first select the required class to query, and then the relevant fields to retrieve. The following parameters are available:

- *offset*: This parameter defines the start position for returning values in the result data set of the query. Default value: 0.
- *limit*: Defines the maximum number (1-1000) of records that are returned for each invocation. Default value: 20.
- *where*: Optional SQL condition for a WHERE clause statement. To use this parameter, use SQL syntax, and if required, the actual column and table names as defined in the Alfabet database.

You can either provide a parameter value at run time or specify a default value in the **Parameters** tab. The value given at run time always takes precedence over the default value.

Note: The fixed query parameter is populated automatically by the Alfabet connector and cannot be overwritten.

Resource Definitions

wmAlfabetConnector_vXX.doctypes:selectObjectResponse

This resource invokes an HTTP GET method against the *select* endpoint of Alfabet.

Create

This resource creates one or more Objects of a pre-selected Class in the Alfabet Database. When defining the Cloud Connector Service, first select the required class to create. Then select the properties of the class to set upon create. Some properties might be marked by Alfabet as mandatory and will be selected automatically and marked as required.

Note: Properties that are references to other objects (*Reference* or *ReferenceArray* types) can be set using this resource and does not require separate invocations of the [create relation](#) resource. The expected value for such properties is the REFSTR property of the Alfabet objects being referred.

Multiple objects of the same class can be created in one invocation.

Resource Definitions

wmAlfabetConnector_vXX.doctypes:createResponse

This resource invokes an HTTP POST method against the *update* endpoint of Alfabet. It uses the *create* operation and possibly a number of *create relation* operations if there are reference properties with values.

Delete

This resource deletes Objects from the Alfabet Database.

The required input for this resource is the REFSTR of the Objects to delete.

Multiple objects can be deleted in one invocation.

Resource Definitions

wmAlfabetConnector_vXX.doctypes:deleteObjectRequest

wmAlfabetConnector_vXX.doctypes:updateResponse

This resource invokes an HTTP POST method against the *update* endpoint of Alfabet, and uses the *delete* operation.

Update

This resource updates scalar properties of an Object of a pre-selected Class in the Alfabet Database. When defining the Cloud Connector Service, first select the required class to update. Then select the properties of the class to update.

Note: Properties that are references to other objects (*Reference* or *ReferenceArray* types) cannot be set using this resource, and will require separate invocations of *create relation* or *delete relation*.

Multiple objects of the same class can be updated in one invocation.

To update an object, provide the REFSTR property.

Resource Definitions

wmAlfabetConnector_vXX.doctypes:updateResponse

This resource invokes an HTTP POST method against the *update* endpoint of Alfabet, and uses the *update* operation.

Create Relation

This resource creates a relation between two Objects in the Alfabet Database.

The required inputs are:

- *_fromref* - REFSTR of the source object.
- *_toref* - REFSTR of the target object.
- *property* - Name of the source object class property establishing the relation.

Multiple relations can be created in one invocation.

Resource Definitions

wmAlfabetConnector_vXX.doctypes:updateRelationRequest

wmAlfabetConnector_vXX.doctypes:updateResponse

This resource invokes an HTTP POST method against the *update* endpoint of Alfabet, and uses the *create relation* operation.

Delete Relation

This resource deletes an existing relation between two Objects in the Alfabet Database.

The required inputs are:

- *_fromref* - REFSTR of the source object.
- *_toref* - REFSTR of the target object.
- *property* - Name of the source object class property establishing the relation.

Multiple relations can be deleted in one invocation.

Resource Definitions

wmAlfabetConnector_vXX.doctypes:updateRelationRequest

wmAlfabetConnector_vXX.doctypes:updateResponse

This resource invokes an HTTP POST method against the *update* endpoint of Alfabet, and uses the *delete relation* operation.

9 FAQ and Troubleshooting

I come across network connectivity related issues, for example, IO and read timeout, while enabling a connection or executing a service.

These errors may be due to network connectivity issues, which can be handled by observing the following guidelines:

- If your company has the proxy server configured, set up the proxy settings in the **Integration Server Administrator**. Go to **Settings > Proxy Servers** and specify the proxy alias in the respective Connector Connection page.
If you have already configured that alias as the default proxy for **Integration Server**, you do not need to specify the proxy alias in the Connector Connection page.
If you have not configured any proxy alias as the default proxy, then you must explicitly set the proxy alias name in the CloudStreams connection page in the **Proxy Server Alias** field.
- In case the network is slow or the backend processing takes longer than usual, increase the Connection Timeout and the Socket Read Timeout values.

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.

© 2016 Software AG. All rights reserved. Software AG and all Software AG products are either trademarks or registered trademarks of Software AG. Other product and company names mentioned herein may be the trademarks of their respective owners.

