WEBMETHODS CLOUDSTREAMS PROVIDER FOR SALESFORCE.COM

Version 10.7 | May 2022

CONTENTS

1 Document Change History

Document revision date	Summary of changes
May 2022	First release of this document.

2 About this Guide

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

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

- Salesforce objects that you want to manage
- Salesforce.com workflow and workflow configurations
- 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 Salesforce.com

webMethods CloudStreams Provider for Salesforce.com contains predefined CloudStreams connectors that you use to connect to different versions of the Salesforce.com cloud applications. Using webMethods CloudStreams, you configure the CloudStreams Salesforce.com connectors 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.

4 Managing Cloud Connections

You can create and manage the cloud connections for each CloudStreams connector using Integration Server Administrator.

Note: For more information on the connection details, see *webMethods CloudStreams Connector Concepts Guide.*

4.1 Creating the Salesforce Cloud Connection

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 section in the *webMethods CloudStreams Connector Concepts Guide.*

Use the following parameters to create a connection:

1. For **Basic view**, go to **Integration Server Administrator** > **Solutions** > **CloudStreams** > **Providers**. For example, select **Salesfrce.com** and then select the connector name. Click on **Configure New Connection**. By default, you will be on **Basic view**.

Section	Field	Connection Type	Description
Connection: Details	Package		The package in which to create the connection. You must create the package using Designer before you can specify it using this parameter. For general information about creating and managing packages, see the *Designer Service Development online help*. By default, the connection is created in the Integration Server Default package. **Note:** Software AG recommends that you configure the connection in a user-defined package. The custom package that you create must have a dependency on the WmCloudStreams package.

Section	Field	Connection Type	Description
	Connection Type		Some SaaS application providers offer multiple authentication options to access their back-end APIs (for example, Salesforce CRM v53). The connector developer needs to create a connector for each authentication scheme. For example, one connector for basic authentication and another connector for OAuth 2.0 authentication and another connector for OAuth 2.0 JWT. From this release on, a connector developer can build a single connector that supports multiple authentication schemes. You can then select different connection types for the same connector and for the same back end from the connection configuration page in Integration Server Administrator.
		Connection Type: Salesforce SOAP connection	Connection for the Salesforce back end using the basic authentication mechanism. Basic authentication is the default authentication type selected.
		Connection Type: Salesforce Web Server Authentication Flow	Connection for the Salesforce back end using OAuth V2.0 authentication mechanism.
		Connection Type: Salesforce OAuth 2.0 JWT Token Flow	Connection for the Salesforce back end using OAuth JWT Bearer Token authentication mechanism.

Section	Field	Connection Type	Description
	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 Designer. For more information about the use of special characters in package, folder, and element names, see the *Designer Service Development online help*.
Connection Groups: Connection	Server URL	Connection Type: Salesforce SOAP connection	The login endpoint to initiate communication with the SaaS provider. For example, for Salesforce.com REST connector version 53, the endpoint URL is https://login.salesforce.com/services/Soap/u/53.0
		Connection Type: Salesforce Web Server authentication Flow	The login endpoint differs based on Connection Type. For example, for Salesforce.com CRM connector version 53 supporting Web Server Authentication Flow, the endpoint URL is https://<instance >/services/Soap/u/53.0 **Note:** Replace <instance> placeholder in the URL with your Salesforce instance.
		Connection Type: Salesforce OAuth 2.0 JWT Token Flow	The login endpoint differs based on Connection Type. For example, for Salesforce.com CRM connector version 53, supporting OAuth 2.0 JWT Token Flow, the endpoint URL is https://<instance >/services/Soap/u/53.0

Section	Field	Connection Type	Description
			Note: Respective endpoint is selected based on Connection Types automatically. Replace <instance> placeholder in the URL with your Salesforce instance. (Applies only if you select OAuth or JWT connection type.)
Connection Groups: Credentials	Connection Type: Salesforce SOAP connection	Username	The name of the user account on the SaaS provider that the connection uses to connect to the SaaS provider. **Note**: When you use SOAP or REST-based Salesforce.com connector, Software AG recommends that you do not use more than one connection with the same username.
		Password	The password for the username provided in the **Username** field. When you access Salesforce.com from outside your company's trusted networks, you must add a security token (provided by Salesforce) to your password. For more information about logging in Salesforce.com, see the Salesforce.com documentation.
	Authorization Type		The type of HTTP authorization scheme to use for the connection. The CloudStreams Provider for Salesforce.com does not use Authorization headers. The username and password are inserted in a login operation that executes when the connection pool is enabled. If you do not specify a value for this field, no additional authorization scheme will be executed at runtime. For example, when you specify a **Username** and **Password** but you do not specify a value for the **Authorization Type**, the user credentials are not inserted into an Authorization header.

Section	Field	Connection Type	Description
			Valid values: basic none ntlm Default: **none**
Connection Groups: OAuth V2.0(Web Sever Authentication Flow)	Connection Type: Salesforce Web Server Authentication Flow	Consumer ID	Consumer Key received from the app. This is a mandatory field.
		Consumer Secret	Consumer Secret received from the app. This is a mandatory field.
		AccessToken	In order to generate an access token, select a link on right top of the Connection Groups 'Generate Access Token', a new window will pop up asking for Consumer Id, Consumer secret. Provide the details to get access token generated.
		Scope	This is a mandatory field, and the value should be, **offline_access api refresh_token**. For more information see in Salesforce org
		Redirect URI	Consumer ID, Consumer Secret, Scope are essential to generate access token, which is needed to get an authorization permission from Salesforce back end to access their API's. Generated access token should be redirected to same page where it is requested. Hence provide the redirected URI same as your Integration Server login page. Click the **Authorize** tab.

Section	Field	Connection Type	Description
Connection Groups: OAuth V2.0 (JWT Token Flow)	Connection Type: Salesforce OAuth V2.0 JWT Token Flow	Issuer	Issuer is the Consumer Key received from the app. This is a mandatory field.
		Subject	Subject species the Username which is used to login SalesForce.com
		Access Token	In order to generate access token, select a link on the right top of the Connection Groups 'Generate Access Token', a new window will pop up asking for Issuer and subject.
		Expiration Time (mins)	The generated access token will be valid till the expiration of the time given. Note: depending on different vendors, the generated access token might be valid post expiration time as well.
		Keystore Name	It is a required field, to get the JWT token generated. Before choosing the Keystore from the pic list, Keystore should be added in Integration Server admin page. In order to add Keystore, please follow below steps: 1. Login to your Integration Server 10.7 2. Select security > select Keystore 3. Click on Create Keystore alias, provide alias name 4. Provide the location of certificate (The JWT bearer flow supports the RSA SHA256 algorithm, which uses an uploaded certificate as the signing secret. The Connected app registers an X509 Certificate for the app).

Section	Field	Connection Type	Description
			For more information about certificate see the documentation of Salesforce.com **Note:** Use the same certificate uploaded in connected apps in Salesforce.com as well in creating a keystore in Integration Server admin Page. 5. Specify the password as ,**password**, retype the same password, and click **Save**. 6. Select the keystore which you added using X509 certificate.
Connection Groups: Transport Protocol	Element Character Set		The encoding to use for the HTTP message components, such as request line and headers. Default: **US-ASCII**
	Strict Transfer Encoding		Indicates 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: Request Headers	Request Header Names		The names of the HTTP request headers to include when sending the login request.
	Request Header Values		The values of the HTTP request headers included in the login request.

Section	Field	Connection Type	Description
			Note: The values in the **Request Header Names** and **Request Header Values** are comma-delimited. Each comma-delimited value in the Request Header Names fields should be mapped to a corresponding comma-delimited value in the **Request Header Values** field. For example: **Request Header Names:** Content-Type, SOAPAction **Request Header Values:** text/xml, login In the example, the header name **Content-Type** has a value **of text/xml**, and the **SOAPAction** name has the value **login**.
Connection Management Properties	Enable Connection Pooling		Indicates whether connection pooling is enabled for a connection. Valid values: **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 always remain in the connection pool, if connection pooling is enabled. When the connector creates the pool, it creates this number of connections. Default: **1**

Section	Field	Connection Type	Description
	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 reuses any inactive connections in the pool, or, if all connections are active, it waits for a connection to become available. Default:**10**
	Pool Increment Size		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. Default:**1**

Section	Field	Connection Type	Description
	Block Timeout (msec)		The number of milliseconds that Integration Server waits to obtain a connection with the SaaS provider before the connection times out and returns an error. For example, you have a pool with **Maximum Pool Size** of 20. If you receive 30 simultaneous requests for a connection, 10 requests will wait for a connection from the pool. If you set the **Block Timeout** to 5000, the 10 requests 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. If you set the **Block Timeout** value too high, you might encounter problems during errors occur. If a request contains errors that delay the response, other requests are not sent. This setting should be tuned in conjunction with the **Maximum Pool Size** to accommodate such bursts in processing. Default: **1000**

Section	Field	Connection Type	Description
	Expire Timeout (msec)		If connection pooling is enabled, the number of milliseconds that an inactive connection can remain in the pool before it is closed and removed from the pool. The connection pool removes 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. This setting should be tuned in conjunction with the **Initial Pool Size** to avoid excessive opening and closing of connections during normal processing. Default: **1000**
	Startup Retry Count		If the initial attempt fails, the number of times that the system should attempt to initialize the connection pool at startup. **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. Default: **0** (a single attempt)
	Startup Backoff Timeout (sec)		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. Default: **10**

Section	Field	Connection Type	Description
	Session Management		The type of time-out for the connection session. Select the type of session management that fits the requirements of your SaaS provider backend. Software AG recommends that you set this field to **idle** if you want the CloudStreams server to manage the session. Valid values: **none**: The CloudStreams server does not manage session time-out. 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 does not time-out. The CloudStreams server considers the idle time-out. 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 times out 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** time-out value expires.
	Session Timeout (min)		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 time-out value specified at the SaaS provider backend.
	Session Management	Connection Type: Salesforce Web Server Authentication Flow	Select the fixed in session management and provide time.

2. Select **Advanced view** to complete the following fields in addition to the fields you configured in the **Basic view**:

   For **Advanced view**, go to **Integration Server Administrator** > **Solutions** > **CloudStreams** > **Providers**. For example, select **Salesforce.com** provider and then select the connector name. **Click on Configure New Connection**. Later, select **Advanced View** on the top right corner.

   Important! If you do not want to use **Advanced view**, skip this step and proceed.

Section	Field	Description
Connection Groups: Connection	Connection Timeout	The number of milliseconds a connection waits before canceling its attempt to connect to the resource. If you specify 0, the connection waits indefinitely. **Important!** Software AG recommends that you specify a value other than 0 to avoid using a socket with no time-out. Default: **180000 (For Salesforce API version 42 and higher)** Default: **30000 (For other Salesforce API versions)**
	Socket Read Timeout	The number of milliseconds in which the client must read a response message from the server. If you specify 0, the connection waits indefinitely. **Important!** Software AG recommends that you specify a value other than 0 to avoid using a socket with no time-out. Default: **180000 (For Salesforce API version 42 and higher)** Default: **30000 (For other Salesforce API versions)**

Section	Field	Description
	Use Stale Checking	Indicates whether the connection performs additional processing to test if the socket is still functional each time the socket is used. Valid values: • **true**: The connection tests the socket. • **false**: The connection does not test the socket. **Note:** The additional testing of the socket adds a little performance overhead. Default: **false**
	Connection Retry Count	The number of times the system should attempt to initialize the connection at startup if the initial attempt fails. 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 back end, the system only retries if **Retry on Response Failure** is set to true. **Default: 3 (For Salesforce API version 42 and higher)** Default: 1 **(For other Salesforce API versions)**
	Retry on Response Failure	Indicates whether the system should attempt to resend the request when the response has failed, even though the request was sent successfully. Valid values: • **true**: The system attempts to re-establish the connection. • **false**: The system does not attempt to re-establish the connection. Default: **true (For Salesforce API version 42 and higher)** Default: **false (For other Salesforce API versions)**

Section	Field	Description
	Use TCP NoDelay	Indicates whether to use algorithm to reduce the number of packets that need to be sent over the network. Valid values: • **true**: Do not optimize the bandwidth consumption. • **false**: Use Nagle's algorithm to optimize the socket usage. Default: **false**
	Socket Linger	The number of seconds before a client socket closes. Valid values: • **-1**: Use the Java VM default. • **0**: Close the socket connection immediately. • *n* > **0**: Wait for *n* seconds before closing the socket connection. Default: **-1**
	Socket Buffer Size	The size of the read and write socket buffers in bytes. Default: **8192**
	Socket Reuse Address	Indicates whether the socket will be reused even if it is in the TIME_WAIT state because of a previous socket closure. Valid values: • **true**: The socket is reused. • **false**: The socket is not reused. Default: **false**

Section	Field	Description
	Proxy Server Alias	The alias name of an enabled proxy server configuration on Integration Server that is used to route the connection. **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.
	Trust Store Alias	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. **Note**: Setting the Integration Server watt.security.cert.wmChainVerifier.trustByDefault property to "true" overrides the value in this field. In this case, the server trusts all remote server peer certificates. If you want to use the **Trust Store Alias** field, set the Integration Server watt.security.cert.wmChainVerifier.trustByDefault property to "false".
	Hostname Verifier	The fully qualified classname of the Apache X509HostnameVerifier interface. Default: **org.apache.http.conn.ssl.StrictHostnameVerifier** 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 org.apache.http.conn.ssl.AllowAllHostnameVerifier that disables hostname enforcement.

Section	Field	Description
Connection Groups: Credentials	Preemptive Auth	Indicates whether authentication credentials are included when sending a request. Valid values: • **true**: Basic authentication credentials are included when sending requests. • **false**: Authentication credentials are not included when sending requests. Default: **false** **Note**: When this field is set to "false," no Authorization header is sent with the initial request. However, if the server returns a security challenge, the Authorization header is included when resending the request.
	Domain Name	The name of the security domain used for the connection.
	Keystore Alias	A text identifier for an Integration Server Keystore alias. A keystore file contains the credentials (private key/signed certificate) that a client needs for SSL authentication.
	Client Key Alias	The alias to the private key in the Integration Server Keystore file specified in the **Keystore Alias** field. The outbound connections use this key to send client credentials to a remote server. **Note**: To send the client's identity to a remote server, you must specify values in both the **Keystore Alias** and the **Client Key Alias** fields.
Connection Groups: Transport Protocol	HTTP Content Character Set	The encoding to use for the HTTP request message. Default: **ISO-8859-1**

Section	Field	Description
	HTTP Protocol Version	The version of the HTTP transport protocol that the connection must use. Valid values: • HTTP/0.9 • HTTP/1.0 • HTTP/1.1 Default: **HTTP/1.1**
	User Agent	The name of the client that the connection includes in the HTTP User-Agent request header to identify the origin of the request. Default: **CloudStreams**
	Use Expect Continue	Indicates 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. Valid values: • **true**: Use the Expect/Continue handshake. • **false**: Do not use the Expect/Continue handshake. Default: **false**
	Wait for Continue Time	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. Default: **3000**

Section	Field	Description
	Use Chunking	Indicates whether to use HTTP/1.1 chunking with a chunk size that matches the socket buffer size. Valid values: • **true**: Use HTTP/1.1 chunking. • **false**: Do not use HTTP/1.1 chunking. Default: **false**
	Follow Server Redirects	Indicates whether the connection follows server redirects. Valid values: • **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**

3. 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 section in *webMethods CloudStreams Connector Concepts Guide.*

5 Managing Connector listeners

You can create and additionally configure subscription and event processing for a connector listener using Software AG Designer.

Note: Before you create a connector listener, 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.

5.1 Creating a Connector Listener

You can create a connector listener using Software AG Designer. For information about how to prepare for Creating a Connector Listener section in the *webMethods CloudStreams Connector Concepts Guide.*

Use the following parameters to create a connection:

- **Basic view**: This is the default view. Use it to view the standard connection parameters for a connector listener.
- **Advanced view**: To view additional connection parameters for a connector listener, click the Advanced view link. The View Listeners screen displays the connection parameters for the connector listeners. You need to configure these parameters in order to enable a connector listener.

Section	Field	Description
Streaming	Listener Connection Type	Default: **Bayeux HTTP Long Polling** The listener connection type indicates the streaming protocol and transport being used for the listener instance. The supported values are Bayeux HTTP Long Polling and HTTP Streaming.
	Package Name	The package in which the connector listener resides. You must create the package using Software AG Designer before you can create a connector listener. For general information about creating and managing packages, see the *Designer Service Development online help*. **Note:** Software AG recommends that you configure the connector listener in a user-defined package. The

Section	Field	Description
		custom package that you create must have a dependency on the WmCloudStreams package.
	Connection Alias	The name of the existing connector which was used while creating the connector listener. The connector listener will use the existing connection you created to connect to Salesforce.com. You need not create a new connection to use the streaming capability.
Connection Groups: Bayeux Configuration	Backoff Increment	Default: **1000** The number of milliseconds that the backoff time increments by every time a connection with the Bayeux server fails. CometD attempts to reconnect after the backoff time elapses.
	Max Backoff	Default: **30000** The maximum number of milliseconds of the backoff time after which the backoff time is not further incremented.
	Max Message Size	Default: **1048576** The maximum size in bytes of an incoming event message.
Connection Groups: Bayeux Extension Configuration	Replay options	• **New (-1)** – Connector listener receives new events that are broadcast after enabling the connector listener. • **All** (-2)- Connector listener receives all events, that is, past events that are within the retention window and also new events. • **LAST-RECEIVED** - Connector listener receives all the events after the last received event by CloudStreams within the retention window. • **CUSTOM** - Connector listener receives all the events specified by a replayId computed by a

Section	Field	Description
		custom service. You can select the replay option as per your requirement.
	Retrieve Replay ID Service	The service to be executed to compute the replayIds of the past events and the value of the computedReplayID is used to specify the replayId during subscription.
Connection Groups: HTTP Parameter Configuration	Streaming API Endpoint	The Streaming endpoint to initiate listening to the events. For example, for Salesforce.com Streaming API version , the endpoint URL is https://{Instance}.salesforce.com/cometd/53.0 **Note:** Replace {instance} with your Salesforce.com instance.
Connection Groups: **Error Callback Configuration**	Error Handler Service Name	Salesforce connector provides errorhandler service which decides error recoverability. When a Streaming Provider triggers error events such as *403: Unknown Client or 401: Authentication invalid*, the CloudStreams server engine invokes error handler service and error handler service identifies whether the error is recoverable or not. If the error is recoverable, the action is taken by the CloudStreams server engine to restore the listener to the function. Note: Here, services:listenerErrorHandler handles Error Recovery of Salesforce Provider.
	Callback Service Name	Salesforce connector provides a ***default*** callback service, which logs status of error recovery. Note: 1. Here, services:listenerCallBackService is configured to log the error and error recovery status. 2. This callback service could be ***customized*** if needed.

Section	Field	Description
	Run As User	The username you want Integration Server to use when running the service.

Once you have configured the above connector listener fields, you can enable the connector listener.

6 CloudStreams Salesforce CRM Connector

6.1 Overview

The following sections describe only the basic information you need to design or use the SOAP operations supported with the CloudStreams Salesforce CRM connector.

For detailed information about each SOAP operation, see the Salesforce.com documentation.

6.2 Connector Details

The connector details include:

- **SaaS Provider**: Salesforce.com
- **Connector Name:** Salesforce CRM
- **API Version**: 25, 29, 31, 37, 42 ,44, 48, 51, 53
- **API Type**: SOAP
- **Developer**: Software AG
- **Group**: Salesforce.com
- **CloudStreams Minimum Version Compatibility**: 10.7
- **Provider Package Name**: WmSalesforceProvider

6.3 Manage Salesforce Connections

You can manage the Salesforce connection by enabling the connection pool and session management. Set the **Session Management** field to "fixed". Ensure that the value is set to a number lesser than the time-out (in minutes), which is specified at the Salesforce.com portal.

Once session management is enabled, CloudStreams calls the login sequence behind-the-scenes, to ensure that the session does not expire when it exceeds the time-out limit.

6.4 SOAP Operations

6.4.1 *Input and Output Signature*

At runtime, all outgoing requests and incoming responses are validated based on the input and output signature document types. If validation fails, the connector service fails with the fault output response. All CloudStreams SOAP-based cloud services do not throw a service exception. In case of an error, the cloud service returns the SOAP fault document.

For more information about the input and output signature of each SOAP operation, see the Salesforce.com documentation.

6.4.2 *DateTime Handling*

The Salesforce CRM connector uses the following format for any dateTime datatype: *yyyy-MM-dd'T'HH:mm:ss.SSS'Z'*.

6.4.3 *Supported Operations*

The following sections describe the SOAP operations supported by the Salesforce CRM connector.

6.4.3.1 *Core Operations*

6.4.3.1.1 *Create*

Creates one or more new records for existing business objects. For example, Account. Note that the 'create' operation allows adding records (business objects) of same type.

Usage notes and limitations

- You cannot create multiple sObjects of different types in Create operation. You should use the *CreateMultiple* operation of connector version 42 and higher.

6.4.3.1.2 *CreateMultiple*

'CreateMultiple' operation allows adding one or more records (sObjects/business objects) of different object types. For example, business objects such as Account and Contact records can be created in a single invocation. In the CreateMutiple operation, you can use external ID fields as a foreign key, which allows you to create a record and relate it to another existing record in a single step.

This operation is present in the CloudStreams SOAP Partner API connector of version 42 and higher.

- If the selected sObject has many external IDs, then only one externalId will be used for a single SObject and the parent record ID will be ignored while using the externalId field. For more information, see section 8.4.3.4.
- You must take care of SOAP API Call Limits. For more information, see the Salesforce API documentation.

6.4.3.1.3 *ConvertLead*

Converts a Lead into an Account, a Contact, or optionally into an Opportunity.

6.4.3.1.4 *Delete*

Deletes one or more records for existing business objects.

6.4.3.1.5 *EmptyRecycleBin*

Deletes all sObjects currently in the Recycle Bin.

6.4.3.1.6 *FindDuplicates*

Performs rule-based searches for duplicate records. The input is an array of sObject, each of which specifies the values to search for and the type of object that supplies the duplicate rules. The output identifies the detected duplicates for each object that supplies the duplicate rules.

6.4.3.1.7 *FindDuplicatesByIDs*

Performs rule-based searches for duplicate records. The input is an array of IDs, each of which specifies the records for which to search for duplicates. The output identifies the detected duplicates for each object that supplies the duplicate rules.

6.4.3.1.8 *GetDeleted*

Retrieves IDs of all deleted sObjects.

6.4.3.1.9 *GetUpdated*

Retrieves IDs of all updated sObjects.

6.4.3.1.10 *InvalidateSessions*

Logs out and makes the session IDs invalid.

6.4.3.1.11 *Merge*

Merges and updates a set of sObjects based on the sObject ID.

6.4.3.1.12 *Retrieve*

Retrieves one or more records based on the specified sObject IDs.

6.4.3.1.13 *Search*

Searches for records based on a search string.

Usage notes and limitations

- The CloudStreams SOAP Partner API connector does not support configuring more than one business object.
- When creating a cloud service, the Salesforce CRM connector shows all the fields of the selected sObject in Software AG Designer. However, at design time, you must select only the fields that can be searched.
- You must specify the Salesforce Object Search Language (SOSL) that matches the sObject fields selected in the seachString input parameter. For more information, see the Salesforce.com SOSL documentation.

6.4.3.1.14 *Undelete*

Restores deleted records from the Recycle Bin.

6.4.3.1.15 *Update*

Updates one or more records for existing business objects. For example, Accounts or Contacts. Note that the Update operation allows updating records (business objects) of same type.

Usage notes and limitations

- The Salesforce CRM connector shows all the fields of the selected sObject in the Software AG Designer. However, at design time, you must select only the fields that can be updated.
- The Update operation cannot update records for multiple object types. You can use "UpdateMultiple" operation of connector version 42 and higher.

6.4.3.1.16 *UpdateMultiple*

The UpdateMultiple operation allows updating one or more records (sObjects/business objects) of different object types. For example, business objects such as Account and Contact records can be updated in a single invocation. In the UpdateMultiple operation, you can use external ID fields as a foreign key, which allows you to update a record and relate it to another existing record in a single step.

This operation is present in the CloudStreams SOAP Partner API connector of version 42 and higher.

- If the selected sObject has many external IDs, then only one externalId is used for a single SObject and the parent record ID will be ignored while using the externalId field.
- You must take care of SOAP API Call Limits. For more information, see the Salesforce API documentation.

6.4.3.1.17 *Upsert*

Creates new records and updates existing records. Uses a custom field to determine whether existing records are present.

6.4.3.1.18 *Query*

Executes a query for a selected sObject and returns data that matches the specified criteria. The following predefined parameters are provided:

- **select**: This parameter contains fields of the selected business object and its related object fields (subquery)/functions. Use this parameter when you want to run a custom complex SOQL query (having sub query, aggregate function, and so on). You can pass any string which is a valid SOQL SELECT clause.
- **where**: This parameter can be used to contain a conditional where clause.
- **limit**: This parameter can be used to contain the number of records to limit for the query.
- **filter**: This parameter contains custom filter condition. If specified, it overrides the where and limit parameters.

You can either provide a parameter value at runtime or specify a default value in the **Parameters** tab. The value given at runtime always takes precedence over the default value. However, if the existing default value is of type *fixed default*, overwrite will fail.

6.4.3.1.19 *QueryAll*

Retrieves data from all sObjects, including the deleted objects. The following predefined input parameters are provided:

- **select**: This parameter contains fields of the selected business object and its related object fields (subquery)/functions. Use this parameter when you want to run a custom complex SOQL query (having sub query, aggregate function, and so on). You can pass any string which is a valid SOQL SELECT clause.
- **where**: This parameter can be used to contain a conditional where clause.
- **limit**: This parameter can be used to contain the number of records to limit for the query.
- **filter**: This parameter contains custom filter condition. If specified, it overrides the where and limit parameters.

You can either provide a parameter value at runtime or specify a default value in the **Parameters** tab. The value given at runtime always takes precedence over the default value. However, if the existing default value is of type *fixed default*, overwrite will fail.

6.4.3.2 *Support for Salesforce Complex Queries*

Client applications sometimes need to query Salesforce Object relationships, for example, Child-to-Parent and Parent-to-Child relationships. Salesforce Object Query Language (SOQL) provides syntax to support these types of queries, called relationship queries, against both standard objects and custom objects. Support for Salesforce complex queries (relationship queries) allows all supported filtering clauses.

In earlier releases, the SOQL SELECT statement was very restrictive in nature. From the 9.7 release, you can execute any Salesforce supported SOQL statement and can use either the *filter* parameter or the *where* and *limit* parameters. Both the *select* and *filter* parameters are newly introduced for advanced users who require greater control on the QUERY. The *where* and *limit* parameters are also retained for backward compatibility.

If you select some fields and use the *select* parameter, the already selected fields are overridden with the values you have entered for the *select* parameter.

From the version 9.7 release, the following two SELECT statements are possible:

SELECT ${select} FROM {BusinessObject} [${filter}]

or

SELECT ${select} FROM {BusinessObject} [WHERE ${where}] [LIMIT ${limit}]

Salesforce Object Query Language (SOQL) supports complex queries (relationship queries). Relationship queries are of two types, Parent-to-Child and Child-to-Parent. Both types of queries are supported.

Out of the box, Salesforce connector supports Child-to-Parent relationships. In the Software AG Designer user interface, you can select business objects and its fields. The CloudStreams framework generates the appropriate query for Child-to-Parent relationships. In this kind of query, you do not need to enter the actual SOQL query.

Child-to-Parent relationship query sample

SELECT FirstName, Account.Name from Contact

This query returns the first names of all the contacts in the organization, and for each contact, the account name associated with (parent of) that contact.

Parent-to-Child relationship query sample

SELECT Name, (SELECT Contact.FirstName, Contact.LastName FROM Account.Contacts) FROM Account

This query returns all accounts, and for each account, the first and the last name of each contact associated with (the child of) that account.

Out of the box, Salesforce connector does not support Parent-to-Child relationship queries. This means that the above SOQL query cannot be automatically generated by the CloudStreams framework. These kinds of queries can be supported with custom fields definition support and by specifying the *select* parameter value.

Using the custom fields' definition capability, you can enhance or update the query, queryMore, or queryAll cloud service signatures with user defined fields and document structure. Typically, you will formulate the required Parent-to-Child query and would know the expected output for a given query. Depending on the expected output, you will have to update the custom fields' definition and also ensure that the design time signature matches the runtime query output signature.

6.4.3.2.1 *QueryMore*

Retrieves the next batch of sObjects from a query.

6.4.3.3 *Utility Operations*

6.4.3.3.1 *Get Server Timestamp*

Retrieves the server timestamp.

6.4.3.3.2 *Get User Info*

Returns standard information for the current user.

6.4.3.3.3 *Reset Password*

Changes the password to a system-generated value temporarily.

6.4.3.3.4 *Set Password*

Sets a specified value as your password.

6.4.3.4 *Support for External ID*

You can use external ID fields as a foreign key, which allows you to create a record and relate it to another existing record in a single step instead of first querying the parent record ID.

For example, if you want to relate sObject "Account" to "Contact" while creating or updating "Contact", then you have the following two options:

- Use the Salesforce unique ID
- Use the external ID field of "Account"

You can select only one of these options, that is, either the Salesforce unique ID or external ID. Using both the Salesforce unique ID and the external ID at the same time is not supported and will result in errors.

When you expand a field of type relationship, you must choose at most one child of this field. Among the external IDs present for a given child, you might only select one of the external IDs. Choosing more than one external ID is not supported and will result in errors.

6.4.4 *Additional functionalities of v37*

The following functionalities have been added for Salesforce v37:

6.4.4.1 *New Headers*

The following new headers have been added for some API calls to provide additional functionality:

<table>
<tr><th>New Header</th><th>API Calls</th><th>Description</th></tr>
<tr><td rowspan="5">LocaleOptions</td><td>Create</td><td rowspan="5">Specifies the language of the labels returned. The value must be a valid user locale (language and country), such as de_DEoren_GB.
For more information on locales, see the Language field on the CategoryNodeLocalization object.
Fields
<table>
<tr><th>Element Name</th><th>Type</th><th>Description</th></tr>
<tr><td>language</td><td>string</td><td>Specifies the language of the labels returned. The value must be a valid user locale (language and country), such as de_DEoren_GB.</td></tr>
</table></td></tr>
<tr><td>Delete</td></tr>
<tr><td>Update</td></tr>
<tr><td>Upsert</td></tr>
<tr><td>Merge</td></tr>
</table>

<table>
<tr>
<td rowspan="3">DuplicateRuleHeader</td>
<td>Create</td>
<td rowspan="3">Determines options for using duplicate rules to detect duplicate records. Duplicate rules are part of the Duplicate Management feature.
Fields
<table>
<tr><th>Element Name</th><th>Type</th><th>Description</th></tr>
<tr><td>allowSave</td><td>boolean</td><td>Set to true to save the duplicate record. Set to false to prevent the duplicate record from being saved.</td></tr>
<tr><td>includeRecordDetails</td><td>boolean</td><td>Set to true to get fields and values for records detected as duplicates. Set to false to get only record IDs for records detected as duplicates.</td></tr>
<tr><td>runAsCurrentUser</td><td>boolean</td><td>Set to true to make sure that sharing rules for the current user are enforced when duplicate rules run. Set to false to use the sharing rules specified in the class for the request. If no sharing rules are specified, Apex code runs in system context and sharing rules for the current user are not enforced.</td></tr>
</table>
</td>
</tr>
<tr><td>Update</td></tr>
<tr><td>Upsert</td></tr>
</table>

6.4.4.2 *Error Handling*

An Error contains information about an error that occurred during a create(), merge(), process(), update(), upsert(), delete(), or undelete() call. For more information, see Error Handling.

An Error has the following properties:

- statusCode
- message
- fields
- extendedErrorDetails

If your organization has active duplicate rules and a duplicate is detected, *SaveResult* includes an error with a data type of **DuplicateError**.

6.4.4.2.1 *Duplicate Error*

It contains information about an error that occurred when an attempt was made to save a duplicate record. Use this if your organization has set up duplicate rules, which are part of the Duplicate Management feature.

- You need to create duplicate rules. Use duplicate rules to define what happens when a user tries to save a duplicate record. Perform the procedure as mentioned in the following link: https://help.salesforce.com/apex/HTViewHelpDoc?id=duplicate_rules_create.htm&language=en_US

- When using the service creation wizard for the Create, Delete, Update, Upsert, and Merge operations, choose the document type DuplicateError {urn:partner.soap.sforce.com} for the errors field when you are prompted with the text Configure Data Types of Fields on screen, and then click Finish.

- The default value of includeRecordDetails is false and when it is false, the *any in the mathcRecords is replaced by Id[] (array of id).

- The default value of includeRecordDetails is true and when it is true, the *any in the matchRecords is replaced by ens:Name[]. While selecting the ens:Name, it contains the value for the matching fields.

Field	Details
duplicateResult	Type : DuplicateResult Description: The details of a duplicate rule and duplicate records found by the duplicate rule.
fields	Type: string[] Description: Array of one or more field names. Identifies which fields in the object, if any, are affected by the error condition.

message	Type: string Description: Error message text.
statusCode	Type : StatusCode Description: A code that characterizes the error. The full list of status codes is available in the WSDL file for your organization.

6.4.5 *Additional functionalities of v42*

In addition to v37 functionalities, the following functionalities have been added for Salesforce v42:

6.4.5.1 *New Operations*

New Operation	API Calls	Description
Create Multiple	Create	Allows adding one or more records (objects/business objects) of different object types.
Update Multiple	Update	Allows updating one or more records (objects/business objects) of different object types.
Find Duplicates	findDuplicates	Performs rule-based searches for duplicate records.
Find Duplicates By Ids	findDuplicatesByIds	Performs rule-based searches for duplicate records based on array of Ids.

6.4.6 *Additional Functionalities of v44*

In addition to v42 functionality, webMethods CloudStreams Provider for Salesforce.com supports Streaming APIs in v44. Streaming API support is bundled with the Salesforce CRM v44 connector.

6.4.6.1 *Events supported in Salesforce CRM v44 connector*

No.	Listener Name	Subscription Path	Description

1	**Salesforce Push Topic Event**	/topic/{pushTopicName}	Listens to incoming streaming notifications for changes to Salesforce data that match a SOQL query you define in PushTopic.
2	**Salesforce Platform Event**	/event/{eventName}	Listens to incoming platform event notifications for changes to Salesforce data that match a SOQL query you define in a secure and scalable way.
3	**Salesforce Change Data Capture Event**	/data/{ChangeEvents}	Listens to change data capture events, which represent changes to Salesforce records that you have subscribed to listen.
4	**Salesforce Generic Event**	/u/notifications/ {GenericStreamingChannel}	Listens to generic events for custom notifications that are not tied to Salesforce data changes.

6.4.6.2 *Cluster support*

The connector listeners, based on the Salesforce CRM connector, work out of the box in multi-node mode in a cluster. In this mode, the processing load is distributed across each cluster node where the connector listener is available in an enabled state, and checks are performed in order to eliminate duplicate processing of events. See the *Administering webMethods CloudStreams* document for more details around clustering support for connector listeners.

6.4.6.3 *Duplicate event check*

Salesforce sends channel and replayId fields in the subscribed events. webMethods CloudStreams Provider for Salesforce.com uses these values to check for duplicate events. The guaranteed events are processed exactly once. You can configure the processing of the subscribed events by adding an Action to be applied on the incoming events. You can log or invoke a service-based on your configurations.

6.4.7 *Additional Functionalities of v48*

All the operations from connector version 44 are supported in version 48.

6.4.8 *Additional Functionalities of v51*

All the operations from connector version 48 are supported in version 51.

6.4.9 *Additional Functionalities of v53*

All the operations from connector version 51 are supported in version 53.

7 CloudStreams Salesforce Bulk Data Loader Connector

7.1 Overview

The CloudStreams Salesforce Bulk Data Loader connector supports REST resources for processing large sets of data.

The following sections describe only the basic information you need to use the REST resources supported with the CloudStreams Salesforce Bulk Data Loader connector.

For detailed information about each REST resource, see the Salesforce.com documentation.

7.2 Connector Details

The connector details include:

- **SaaS Provider**: Salesforce.com
- **Connector Name**: Salesforce Bulk Data Loader
- **API Version**: 25, 29, 31, 37, 42,44, 48, 51, 53
- **API Type**: REST
- **Developer**: Software AG
- **Group**: Salesforce.com
- **CloudStreams Minimum Version Compatibility**: 10.7
- **Provider Package Name**: WmSalesforceProvider

7.3 Manage Salesforce Connections

You can manage the Salesforce connection by enabling the connection pool and session management. Set the **Session Management** field to "fixed". Ensure that the value is set to a number lesser than the time-out (in minutes), which is specified at the Salesforce.com portal.

Once session management is enabled, CloudStreams calls the login sequence behind-the-scenes, to ensure that the session does not expire when it exceeds the time-out limit.

7.4 REST Resources

7.4.1 *Request and Response Processing*

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

When creating a REST connector service in Designer, if the processing type in one of the fields in the **Request Processing** or **Response Processing** sections is set to **Document**, do *not* set the processing type in the other field to **Binary Stream**. When you set the processing type to **Document**, you must also choose an appropriate application/*n* type, where *n* is the format of the response message, for example, application/*xml*.

For more information about the supported processing types, see the Salesforce.com documentation.

7.4.2 *Input and Output Signature*

The CloudStreams Bulk API connector contains an Integration Server document that maps to the request or response data for a REST resource, for example, *wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_JobInfo*, where *XX* is the version of the Salesforce Bulk Data Loader connector.

The input values to the fields in the service signature map directly to the input values required by the Salesforce Bulk Data Loaders. For information about the required fields to configure for a request, see the Salesforce.com documentation.

For stream-based request input values, no additional encoding is applied to the request stream. You must ensure that the stream is encoded to match the requirements of the backend.

7.4.3 *Error and Fault Handling*

For most of the REST resources, the output signature refers to an Integration Server document when the response is successful and to an error document when the response fails. For example, when the CloudStreams server returns the HTTP error code 400 response for a bad request, the output signature refers to an error document.

When the Salesforce.com backend returns an error code that cannot be mapped, for example HTTP error code 404 for a wrong server URL, the Salesforce Bulk Data Loader connector returns a fault document with details about the error response. Also, the Salesforce Bulk Data Loader connector does not throw a ServiceException if a processing failure occurs. The Salesforce Bulk Data Loader connector returns a fault document if the processing fails for any reason during an outbound request or inbound response. You can view the fault document in the service execution output pipeline in Designer.

7.4.4 *DateTime Handling*

The Salesforce Bulk Data Loader connector uses the following format for any dateTime datatype: *yyyy-MM-dd'T'HH:mm:ss.SSS'Z'*.

7.4.5 *Usage Notes*

The CloudStreams server uses the Salesforce Bulk Data Loader connector XML schemas to locate the available resources and their respective Integration Server document types. For example, *wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_JobInfo*.

The Salesforce Bulk Data Loader connector uses the XML schemas to generate Integration Server documents that have all the required resource definitions for a given resource. The Integration Server documents are mapped and used by a specific REST resource to send requests and receive responses.

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

7.4.6 *Supported Resources*

The following sections describe the REST resources supported by the Salesforce Bulk Data Loader connector. For detailed information about the REST resources, see the Salesforce.com documentation.

7.4.6.1 *CreateJob*

Creates a new job by sending a POST request to the designated URI.

Resource Definitions

wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_JobInfo

Usage Notes

While creating a job, you must select the *tns:contentType* of the **CreateJob** resource, keeping in mind the type of the batch to be submitted for this job.

7.4.6.2 *CreateBatch*

Adds a new batch to a job by sending a POST request to the designated URI.

Resource Definitions

wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_BatchInfo

Usage Notes

The header *Content-Type* value that the **CreateBatch** resource connector service uses should match the expected job and response data type. For example, for a batch that submits XML data, the *tns:contentType* definition of the **CreateJob** resource should be set to *XML* and the header *Content-Type* for the corresponding **CreateBatch** resource connector service should be set to *application/xml*. For a batch that submits CSV data, the *tns:contentType* definition of the **CreateJob**

resource should be set to *CSV* and the header *Content-Type* for the corresponding **CreateBatch** resource connector service should be set to *text/csv*.

7.4.6.3 *Job*

Aborts or closes an existing job.

Resource Definitions

wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_JobInfo

7.4.6.4 *JobDetails*

Retrieves the details of an existing job.

Resource Definitions

wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_JobInfo

7.4.6.5 *BatchInfoList*

Returns the list of batches submitted for a given job ID.

Resource Definitions

wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_BatchInfoList

7.4.6.6 *BatchStatus*

Provides details of a submitted batch, for example, state.

Resource Definitions

wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_BatchInfo

7.4.6.7 *BatchResult*

Retrieves the Batch Result List.

Resource Definitions

The service signature is mapped to the *wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_BatchResult* Integration Server document. This Integration Server document receives the response for all types of submitted batches, except query batches. For the results of query batches, see the **ResultList** resource. If you use the **BatchResult** resource to retrieve the results of a submitted query batch, the cloud service fails with a validation error.

7.4.6.8 *ResultList*

Retrieves the results of a query batch.

Resource Definitions

The service signature is mapped to the *wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_ResultList* Integration Server document. Use the **ResultList** resource only for submitted query batches. If you use the **ResultList** resource for other types of batches, the cloud service fails with a validation error.

7.4.6.9 *QueryResult*

Retrieves the result of a query batch.

Resource Definitions

None.

The response will be a stream and will depend on the submitted query.

7.4.6.10 *GetBatchRequest*

Retrieves the request details of a batch, for example, the request content submitted for a batch.

Resource Definitions

None.

The response is a stream and will depend on the submitted batch request.

8 Salesforce Analytics Cloud Connectivity

For Salesforce Analytics Cloud connectivity, note the following points:

- The External Data API related business objects are available only from Salesforce connector version 31.
- If you want to integrate external data into Analytics Cloud, you must select the following business objects:
  - External Data
  - External Data Part

  These objects are referred to as **InsightsExternalData** and **InsightsExternalDataPart** respectively in the Salesforce documentation.
- For the **External Data Part** business object, pass the **DataFile** field as a base64 encoded string value.