webMethods CloudStreams Provider for Salesforce.com Installation and User’s Guide

Version 9.6

April 2014

This document applies to webMethods CloudStreams Provider for Salesforce.com Version 9.6 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: WST-XCS-IUG-96-20140415

Table of Contents

About this Guide......7
- Document Conventions......7
- Documentation Installation......8
- Online Information......8

What Is webMethods CloudStreams Provider for Salesforce.com?......11
- What Is a CloudStreams Connector?......12
- What Is a Cloud Connection?......12
  - Connection Pools......13
  - Run-Time Behavior of Connection Pools......13
- What Is a Cloud Connector Service?......13

Installing and Uninstalling CloudStreams Provider for Salesforce.com......15
- Requirements......16
- Installing CloudStreams Provider for Salesforce.com 9.6......16
- Uninstalling webMethods CloudStreams Provider for Salesforce.com 9.6......17

Managing CloudStreams Connectors......19
- Enabling CloudStreams Connectors......20
- Disabling CloudStreams Connectors......20
- Deleting CloudStreams Connectors......21
- Configuring the Properties of CloudStreams Connectors......21

Managing Cloud Connections......23
- Creating Cloud Connections......24
- Enabling Cloud Connections......37
- Viewing Cloud Connections......38
  - Sorting and Filtering Connections......38
- Disabling Cloud Connections......39
- Editing Cloud Connections......39
- Dynamically Changing a Cloud Service's Connection at Run Time......40
- Copying Cloud Connections......40
- Deleting Cloud Connections......41

Managing Cloud Connector Services......43
- Creating a Cloud Connector Service......44
- Editing a Cloud Connector Service for a SOAP-Based Provider......45
- Editing a Cloud Connector Service for a REST-Based Provider......48
  - Viewing the Constraints Applied to Variables......54

CloudStreams Salesforce.com Partner SOAP API Connector......55
- Overview......56
- SOAP Cloud Connections......56

Designing SOAP Operations....56
SOAP Operations....57
Input and Output Signature....57
DateTime Handling....57
Supported Operations....57
Core Operations....57
Create....57
ConvertLead....58
Delete....58
EmptyRecycleBin....58
GetDeleted....58
GetUpdated....58
InvalidateSessions....58
Merge....58
Retrieve....58
Search....58
Undelete....58
Update....59
Upsert....59
Query....59
QueryAll....59
QueryMore....59
Utility Operations....60
Get Server Timestamp....60
Get User Info....60
Reset Password....60
Set Password....60

CloudStreams Salesforce.com Bulk API Connector....61
Overview....62
REST Cloud Connections....62
REST Resources....62
Request and Response Processing....62
Input and Output Signature....63
Error and Fault Handling....63
DateTime Handling....63
Usage Notes....63
Supported Resources....64
CreateJob....64
CreateBatch....64
Job....64
JobDetails....65
BatchInfoList....65
BatchStatus....65
BatchResult....65

ResultList.. 65
QueryResult..66
GetBatchRequest.. 66

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

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies storage locations for services on webMethods Integration Server, using the convention *folder.subfolder:service* .
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
Italic	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
`Monospace font`	Identifies text you must type or messages displayed by the system.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.

Convention	Description
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Documentation Installation

You can download the product documentation using the Software AG Installer. The documentation is downloaded to a central directory named _documentation in the main installation directory (SoftwareAG by default).

Online Information

You can find additional information about Software AG products at the locations listed below.

If you want to...	Go to...
Access the latest version of product documentation.	**Software AG Documentation website** http://documentation.softwareag.com
Find information about product releases and tools that you can use to resolve problems. See the Knowledge Center to: ■ Read technical articles and papers. ■ Download fixes and service packs (9.0 SP1 and earlier). ■ Learn about critical alerts. See the Products area to: ■ Download products. ■ Download certified samples. ■ Get information about product availability.	**Empower Product Support website** https://empower.softwareag.com

If you want to...	Go to...
■ Access older versions of product documentation. ■ Submit feature/enhancement requests.	
■ Access additional articles, demos, and tutorials. ■ Obtain technical information, useful resources, and online discussion forums, moderated by Software AG professionals, to help you do more with Software AG technology. ■ Use the online discussion forums to exchange best practices and chat with other experts. ■ Expand your knowledge about product documentation, code samples, articles, online seminars, and tutorials. ■ Link to external websites that discuss open standards and many web technology topics. ■ See how other customers are streamlining their operations with technology from Software AG.	**Software AG Developer Community for webMethods** http://communities.softwareag.com/

1 What Is webMethods CloudStreams Provider for Salesforce.com?

- What Is a CloudStreams Connector? 12
- What Is a Cloud Connection? 12
- What Is a Cloud Connector Service? 13

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 *Administering webMethods CloudStreams*.

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 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" on page 19.

What Is a Cloud Connection?

You 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 configure connections using Integration Server Administrator. For a list of tasks you must do before creating connections, see "Managing Cloud Connections" on page 23. For more information about configuring cloud connections, see "Creating Cloud Connections" on page 24.

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.

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" on page 24.

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 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" on page 44.
2. Edit the cloud connector service using the service editor in 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 SOAP-Based Provider" on page 45 and "Editing a Cloud Connector Service for a REST-Based Provider" on page 48.

For more information about each type of CloudStreams connector requests, see:

- "CloudStreams Salesforce.com Partner SOAP API Connector" on page 55.
- "CloudStreams Salesforce.com Bulk API Connector" on page 61.

2 Installing and Uninstalling CloudStreams Provider for Salesforce.com

- Requirements 16
- Installing CloudStreams Provider for Salesforce.com 9.6 16
- Uninstalling webMethods CloudStreams Provider for Salesforce.com 9.6 17

The following sections explain how to install and uninstall webMethods CloudStreams Provider for Salesforce.com 9.6. The instructions use the Software AG Installer and Software AG Uninstaller wizards. For complete information about the wizards or other installation methods, or to install other webMethods products, see the webMethods installation guide for your release. See “About this Guide” for specific document titles.

Requirements

For a list of the operating systems and webMethods products supported by the CloudStreams Provider for Saleforce.com, see *webMethods System Requirements*, available in the webMethods area of the Software AG Documentation Web page.

CloudStreams Provider for Salesforce.com 9.6 has no hardware requirements beyond those of its host Integration Server.

Installing CloudStreams Provider for Salesforce.com 9.6

To install CloudStreams Provider for Salesforce.com 9.6

1. Download Software AG Installer from the Empower Product Support website.
2. Start the Installer wizard.
3. Choose the webMethods 9.6 release. The release includes the webMethods CloudStreams framework and the Integration Server on which to install the provider.
4. Specify the installation directory to use.

   The Installer will install the provider in the *Integration Server_directory*\instances \instance_name \packages directory.
5. In the product selection list, select **CloudStreams > Providers > Salesforce.com Provider 9.6**.

   To install documentation for CloudStreams Provider for Salesforce.com, on the Documentation panel in Installer, select **webMethods Readmes and Documentation**.

   Alternatively, you can download the documentation later from the Software AG Documentation Web page.
6. Start the host Integration Server.

Uninstalling webMethods CloudStreams Provider for Salesforce.com 9.6

To uninstall CloudStreams Provider for Salesforce.com 9.6

1. Shut down the host Integration Server. You do not need to shut down any other webMethods products or applications that are running on your machine.
2. Start Software AG Uninstaller, selecting the webMethods installation directory that contains the host Integration Server. In the product selection list, select **CloudStreams > Providers > Salesforce.com Provider 9.6**. You can also choose to uninstall documentation.
3. After Uninstaller completes, restart the host Integration Server.

Uninstaller removes all CloudStreams Provider for Salesforce.com 9.6-related files that were installed. However, it does not delete files created after you installed the provider (for example, user-created or configuration files), nor does it delete the provider directory structure. You can go to the *Integration Server_directory*\instances \instance_name \packages directory and delete the WmSalesforceProvider directory.

3 Managing CloudStreams Connectors

- Enabling CloudStreams Connectors 20
- Disabling CloudStreams Connectors 20
- Deleting CloudStreams Connectors 21
- Configuring the Properties of CloudStreams Connectors 21

You 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 Salesforce.com on the same machine. For details, see *Installing webMethods and Intelligent Business Operations Products* and "Installing and Uninstalling CloudStreams Provider for Salesforce.com" on page 15.
2. Make sure 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, make sure the cloud 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 connectors, if you have not already done so. See Designer Service Development online help for instructions.

Enabling CloudStreams Connectors

Note: By default, all CloudStreams connectors are enabled.

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 (e.g., Salesforce.com).
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.

Disabling CloudStreams Connectors

You 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 (e.g., Salesforce.com).
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.

Deleting CloudStreams Connectors

You can delete CloudStreams connectors using Integration Server Administrator.

> **Important:** If you delete a connector, you can restore it only by re-installing the connector as described in "Installing CloudStreams Provider for Salesforce.com 9.6" on page 16.

To delete a cloud connector

1. In Integration Server Administrator, go to **Solutions > CloudStreams > Providers**.
2. Click the name of your cloud provider (e.g., Salesforce.com).
3. On the Connectors screen, make sure 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.

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, Salesforce.com.
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	Hides/shows the document type packages that CloudStreams generated from the provider's WSDL or XSD. ■ **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.

4 Managing Cloud Connections

- Creating Cloud Connections 24
- Enabling Cloud Connections 37
- Viewing Cloud Connections 38
- Disabling Cloud Connections 39
- Editing Cloud Connections 39
- Dynamically Changing a Cloud Service's Connection at Run Time 40
- Copying Cloud Connections 40
- Deleting Cloud Connections 41

You 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 *Installing webMethods and Intelligent Business Operations Products* and "Installing CloudStreams Provider for Salesforce.com 9.6" on page 16.
2. Make sure you have webMethods administrator privileges so that you can access the webMethods CloudStreams connector administrative screens. For information about setting user privileges, see *Administering Integration Server*.
3. Start Integration Server and Integration Server Administrator if they are not already running.
4. Using Integration Server Administrator, make sure 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.

Creating Cloud Connections

You 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" on page 23.

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, Salesforce.com.
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, optional, parameters for a cloud connection. To access the Advanced view, click the **Advanced view** link.
6. On the Connections page, complete the following fields:

Section	Field	Description
	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.
	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**	The login endpoint to initiate communication with the SaaS provider. For example, for Salesforce.com REST connector version 25, the end point URL is https://

Section	Field	Description
		login.salesforce.com/services/Soap/u/25.0
Connection Groups: Credentials	**Username**	The name of the user account on the SaaS provider that the connection will use to connect to the SaaS provider. **Note:** When you use a SOAP or REST-based Salesforce.com connector, Software AG does not recommend using more than one connection with the same user name.
	Password	The password for the user name 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 on 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 user name 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 run time. For example, when you specify a **Username** and **Password**, but you do not specify a value for **Authorization Type**, the user credentials are not inserted into an Authorization header. Valid values:

Section	Field	Description
		■ **basic** ■ **none** Default: **none**
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	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.
	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 a value of **login**.	

Section	Field	Description
Connection Management Properties	**Enable Connection Pooling**	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 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**
	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**
	Block Timeout	The number of milliseconds that Integration Server will wait to obtain a connection with the SaaS

Section	Field	Description
		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 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. 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. Default: **1000**
	Expire Timeout	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. 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. This setting should be tuned in conjunction with the **Initial Pool Size** to avoid excessive opening/closing

Section	Field	Description
		of connections during normal processing. Default: **1000**
	Startup Retry Count	The number of times that the system should attempt to initialize the connection pool at startup if the initial attempt fails. **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 Secs	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**
	Session Management	The type of timeout for the connection session. Select the type of session management that fits the requirements of your SaaS provider backend. Software AG recommends setting this field to **idle** if you want the CloudStreams server to manage the session. Valid values: - **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

Section	Field	Description
		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	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.

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

   Important: If you do not want to use the Advanced view, skip this step and continue with step 8.

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 specifying a value other than `0` to avoid using a socket with no timeout. Default: **30000**

Section	Field	Description
	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 specifying a value other than `0` to avoid using a socket with no timeout. Default: **30000**
	Use Stale Checking	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 backend, the system will only retry if **Retry on Response Failure** is set to true. Default: **1**
	Retry on Response Failure	Whether the system should attempt to resend the request when the response has failed, even though the request was sent successfully. Valid values:

Section	Field	Description
		■ **true** The system attempts to re-establish the connection. ■ **false** The system does not attempt to re-establish the connection. Default: **false**
	Use TCP NoDelay	Whether the connection uses the Nagle's algorithm to optimize socket usage. Valid values: ■ **true** The connection does not use the Nagle's algorithm. ■ **false** The connection uses the Nagle's algorithm. 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 *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	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 will be reused. ■ **false** The socket will not be reused. Default: **false**

Section	Field	Description
	Proxy Server Alias	The alias name of an enabled proxy server configuration on Integration Server that will be 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 will trust 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**	Whether authentication credentials are included when sending a request. Valid values: - **true** Basic authentication credentials are included when sending requests. - **false** No authentication credentials are 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 re-sending 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 will 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	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**
	Use Chunking	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.

Section	Field	Description
		■ **false** Do not use HTTP/1.1 chunking. Default: **false**
	Follow Server Redirects	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**

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" on page 37.

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, Salesforce.com.
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 adapter connection and displays **Yes** in the **Enabled** column.

Viewing Cloud Connections

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

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

To view 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, Salesforce.com.
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" on page 24.

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 **Submit**. The Connections screen displays the connections that match the filter criteria.

d. To re-display all connections, click **Show All Connections**.

The Connections screen appears, listing all the current connections.

Disabling Cloud Connections

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

To disable 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, Salesforce.com.
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.

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

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 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, Salesforce.com.
3. On the Connectors screen, click the Connector Name for the CloudStreams connector whose connection you want to edit.
4. Make sure 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" on page 24.

6. Click **Save Changes**.

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.

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 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, Salesforce.com.
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. 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" on page 24.

5. Click **Create**.

Deleting Cloud Connections

You delete cloud connections using Integration Server Administrator.

To delete 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, Salesforce.com.
3. On the Connectors screen, click the Connector Name for the CloudStreams connector whose connection you want to delete.
4. Make sure 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.

5 Managing Cloud Connector Services

- Creating a Cloud Connector Service 44
- Editing a Cloud Connector Service for a SOAP-Based Provider 45
- Editing a Cloud Connector Service for a REST-Based Provider 48

You create and manage cloud connector services using 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.
- If you are working with a SOAP-based provider, you should 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 more information about SOAP operations, see "CloudStreams Salesforce.com Partner SOAP API Connector" on page 55.
- If you are working with a REST-based provider, you should create at least one cloud connector service for each REST resource. For more information about REST resources, see "CloudStreams Salesforce.com Bulk API Connector" on page 61.

Creating a Cloud Connector Service

You create a cloud connector service using Designer.

To create a cloud connector service

1. Open the Service Development perspective in 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**.

   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. Click **Next**.
4. On the Connector page of the wizard, select the CloudStreams Connector associated with the cloud application provider you want to access. 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. 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**.

   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...
SOAP-based	"Editing a Cloud Connector Service for a SOAP-Based Provider" on page 45
REST-based	"Editing a Cloud Connector Service for a REST-Based Provider" on page 48

Editing a Cloud Connector Service for a SOAP-Based Provider

Editing a cloud connector service for a SOAP-based provider consists of specifying the operation, the business object associated with the operation, 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. You edit a cloud connector service using the service editor in 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" on page 44.
- webMethods CloudStreams provides a default connector virtual service for policy enforcements, called WmCloudStreams.SoapVS. 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 *Administering webMethods CloudStreams*.
- 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 SOAP-based provider

1. Open Designer if it is not already open.
2. Navigate to and open the cloud connector service you created in "Creating a Cloud Connector Service" on page 44.

   The service opens in the cloud connector service editor.

3. On the Operation tab, from the **Connector Virtual Service** list, select the connector virtual service to be used for policy enforcement.

   For more information about CloudStreams connector virtual services, see *Administering webMethods CloudStreams*.

4. To configure the operation, business object, fields, and data types of fields, click ✎ next to **Operation**. Designer displays the Operation and Business Object Configuration wizard.

   a. Select the operation you want the cloud connector service to execute, and then click **Next**.

      When you change an operation, Designer clears all the metadata that were associated with the previously selected operation, including the headers, parameters, and data types of fields. You can select the metadata that the updated operation requires in the next steps.

      For more information about SOAP operations, see "CloudStreams Salesforce.com Partner SOAP API Connector" on page 55.

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

   b. In the Select the Business Object page, select a business object and click **Next**.

   c. 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.

      If you want to configure concrete types for the abstract types in the operation you selected, click **Next**. If the operation you selected does not have any abstract type field, click **Finish**.

   d. 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 operation.

   e. Click **Finish**. Designer displays a confirmation message. Click **OK** to update the operation. Designer replaces the existing operation and associated metadata with the updated or default information.

5. On the Headers tab, do the following:

   a. To include a header as part of the service signature, select the **Active** check box next to the header.

   b. To specify a default value for the header variable, click the **Default Value** box next to the variable and type or paste a default value. If the variable is null in the input pipeline, this default value will be used at run time. The value given at run time always take precedence over the default value. However, if the existing default value is of type "fixed default", the overwrite will fail.

c. Repeat the above steps in the Output section of the tab to select the SOAP headers whose contents you want to add to the service's output pipeline.

> **Note:** If the operation you selected on the Operation tab has mandatory headers, Designer displays those headers in gray. You cannot edit or delete a mandatory header.

6. If the operation you selected has predefined input parameters (for example, the Query and QueryAll operations have the `where` and `limit` parameters), you can configure them on the Parameters tab as follows:

   a. To specify a default value for a parameter, click the **Default Value** box next to the parameter. Then, type or paste a default value. If the variable is null in the input pipeline, this default value will be used at run time. The value given at run time always take precedence over the default value. However, if the existing default value is of type "fixed default", the overwrite will fail.

   b. If a predefined parameter is not mandatory, you can activate/de-activate the parameter by clicking the **Active** check box.

   If a predefined parameter is mandatory, Designer displays the parameter in gray and the **Active** check box is selected. You cannot de-activate or delete a mandatory parameter.

   c. To move a parameter up in the list, select the parameter and click ⬆. To move a parameter down in the list, select the parameter and click ⬇.

7. If you want to add other parameters to the service signature, such as variables to be replaced at run time with a user's input, do the following on the Parameters tab:

   a. Click 🗋.

   b. Assign a name to the new parameter. If you want to rename the parameter later, click its name and type a new name.

   c. To specify a default value for the parameter, click the **Default Value** box next to the parameter. Then, type or paste a default value. If the variable is null in the input pipeline, this default value will be used at run time. The value given at run time always take precedence over the default value. However, if the existing default value is of type "fixed default", the overwrite will fail.

   d. You can activate/de-activate the parameter by clicking the **Active** check box, or you can delete it by selecting the parameter and clicking ✖.

   e. To move a parameter up in the list, select the parameter and click ⬆. To move a parameter down in the list, select the parameter and click ⬇.

8. On the Input/Output tab, do the following:

   a. To have the server validate the input to the service against the service input signature, select the **Validate input** check box.

   b. To have the server validate the output to the service against the service output signature, select the **Validate output** check box.

c. Review the service's input and output signature and make any necessary changes as follows:

To change the...	Go to the...
Virtual service name, operation, business object, fields, and data types of fields	Operation tab
List of headers in the requestHeaders or responseHeaders sections, or their default values	Headers tab
List of parameters in the parameters section, or their default values	Parameters tab

The requestBody and responseBody sections are derived from the operation you selected on the Operation tab. The value of $connectionAlias is derived from the connection pool you specified when you first created the cloud connector service. The fault section is derived from the operation response. You cannot change these values in the editor.

9. On the Logged Fields tab, do the following:

   a. Select the check boxes next to the fields you want to log at run time.

   b. If you want to create an alias for a logged field to make it easier to locate in 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 Designer.

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

11. Click **File** > **Save** to save your changes.

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, and default values for parameters included in the input/output signature. You edit a cloud connector service using the service editor in 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" on page 44.

- 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 *Administering webMethods CloudStreams*.
- 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 below.

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

1. Open Designer if it is not already open.
2. Navigate to and open the cloud connector service you created in "Creating a Cloud Connector Service" on page 44.

   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 *Administering webMethods CloudStreams*.

   b. Click ✎ next to **Resource Name**. On the Resource Details dialog box, select the REST resource you want the cloud connector service to process, and then click **OK**.

   On the Resource tab, Designer displays the resource name and description, the HTTP method used to process the REST resource, and the path showing the location in which the resource file resides.

   For more information about REST resources, see "CloudStreams Salesforce.com Bulk API Connector" on page 61.

   c. 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

Option	Meaning
	document type exists or when it is not practical to provide a schema description of the content. For example, the content that is posted for the Salesforce.com “createBatch” resource has a complex structure of fields and rows. A batch of new accounts can be created, and each account can have dozens of fields with precise formatting requirements (for example, date fields). Attachments can even be included in the batch file. The stream option is the best option for this type of resource.

Note: If the resource you selected does not contain any requests, this list is not available.

d. In the **Request Processing** section, select an appropriate serialization type. The serialization type constructs the cloud connector service’s input signature and determines how webMethods CloudStreams should serialize the request data to the cloud provider.

Option	Meaning
application/xml	Formats the request message in XML. Select this option for REST XML service calls that include a message body with the request.
application/x-www-form-urlencoded	Formats the request message using the default HTML Form content type. Select this option for REST XML service calls that do not have a message body.
application/json	Formats the request message using JSON notation.
Stream	Builds the request message as a binary stream.

Note: If the resource you selected has no request message body to send, this list is not available.

e. In the **Response Processing** section, select an appropriate parsing type. The parsing type determines how webMethods CloudStreams takes the response data from the cloud provider.

Option	Meaning
application/xml	Builds the response message in XML. Select this option for REST XML service calls that include a message body with the response.
application/x-www-form-urlencoded	Builds the response message using the default HTML Form content type. Select this option for REST XML service calls that do not have a message body.
application/json	Builds the response message using JSON notation.
Stream	Builds the response message as a binary stream.

Note: If the resource you selected does not contain any responses, this list is not available.

f. 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	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. **Note:** This option works in conjunction with the response's parsing type property. If you select **Stream** as the response's serialization type, Designer also selects **Stream** as the response's parsing type.

> **Note:** If the resource you selected does not contain any responses, this list is not available.

4. On the Headers tab, Designer displays the default HTTP transport headers for the resource, along with their default values. At run time, while processing the headers, webMethods CloudStreams substitutes values as necessary (for example, replaces the “cn.sessionToken” value in the X-SFDC-Session header with the actual runtime session ID). In order to customize the headers, do the following:

   a. 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”, the overwrite will fail as mentioned earlier.

   b. 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.

   c. 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 ⬇.

   d. To include a header as part of the service signature, select the **Active** check box next to the header.

   e. 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 only appear in the pipeline signature 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, Designer displays the configured resource parameters. In order to customize the parameters, do the following:

   a. Review the details about the resource parameters. 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 document *Administering webMethods CloudStreams*.

b. 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”, the 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:

a. To have the server validate the input to the service against the service input signature, select the **Validate input** check box.

b. To have the server validate the output to the service against the service output signature, select the **Validate output** check box.

c. 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:

a. Select the check boxes next to the fields you want to log at run time.

b. If you want to create an alias for a logged field to make it easier to locate in Designer, click the **Alias** box next to a field and type the alias name.

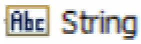
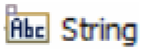
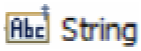
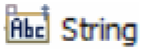
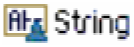
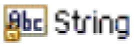
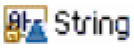
For more information about logged fields, see the section on logging input and output fields in Designer.

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

9. Click **File > Save** to save your changes.

Viewing the Constraints Applied to Variables

Designer displays small symbols next to a variable icon to indicate the constraints applied to the variable. 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.

6 CloudStreams Salesforce.com Partner SOAP API Connector

- Overview 56
- SOAP Cloud Connections 56
- Designing SOAP Operations 56
- SOAP Operations 57

Overview

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

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

SOAP Cloud Connections

When starting a SOAP cloud connection, the Salesforce.com Partner SOAP API connector invokes the Salesforce.com *login* operation. The *login* operation uses the values you specify when configuring a SOAP cloud connection in Integration Server Administrator. When you enable a connection, the CloudStreams server creates the login SOAP request envelope using the input mapping information. The CloudStreams server sends the login SOAP envelope to the Salesforce Server endpoint URL that is specified for the connection. The login response is parsed by applying the output mapping that extracts information, such as the server URL and session ID. The server URL and session ID are persisted in the SOAP cloud connection. The cloud connection is stored in the connection pool.

When stopping a SOAP cloud connection, the Salesforce.com Partner SOAP API connector invokes the Salesforce.com *logout* operation. When disabling the cloud connection, the CloudStreams server creates the logout SOAP request envelope by using the input mapping information. The CloudStreams server sends the logout SOAP envelope to the Salesforce Server URL that is stored in the cloud connection during the login operation. After the logout operation is completed, the associated CloudStreams cloud connection is removed from the connection pool and the entire pool is destroyed.

Designing SOAP Operations

The CloudStreams Salesforce.com Partner SOAP API supports the following types of SOAP operations:

- Login
- Logout
- Metadata
- Simple
- Complex

When designing cloud services with the Salesforce.com Partner SOAP API connector, only operations that are marked as non-hidden are shown in Designer.

With a Complex type operation, you must resolve the abstract SOAP service signature that you obtain initially into a specific input and output signature by selecting the appropriate business object and its associated fields.

All types of operations have the session ID key (*tns:SessionHeader/tns:sessionId*) in the SOAP operation header input. When creating a cloud service, you do not specify the value for the session ID key explicitly. At run time, the CloudStreams server automatically passes the value for the session ID key to the cloud service.

SOAP Operations

Input and Output Signature

At run time, 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.

DateTime Handling

The Salesforce.com Partner SOAP API connector uses the following format for any dateTime datatype: `yyyy-MM-dd'T'HH:mm:ss.SSS'Z'`.

Supported Operations

The following sections describe the SOAP operations supported by the Salesforce.com Partner SOAP API connector.

Core Operations

Create

Creates one or more new records for existing business objects, for example Account.

Usage notes and limitations:

- You cannot create multiple sObjects of different types. You should create a separate service for each business object.
- The CloudStreams SOAP Partner API connector does not support creating a business object using the external ID field.

ConvertLead

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

Delete

Deletes one or more records for existing business objects.

EmptyRecycleBin

Deletes all sObjects currently in the Recycle Bin.

GetDeleted

Retrieves the IDs of all deleted sObjects.

GetUpdated

Retrieves the IDs of all updated sObjects.

InvalidateSessions

Logs out and makes the session IDs invalid.

Merge

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

Retrieve

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

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.com Partner SOAP API connector shows all fields of the selected sObject in Designer. However, at design time you must only select the fields that can be searched.
- You must specify the SOSL (Salesforce Object Search Language) that matches the sObject fields selected in the *seachString* input parameter. For more information, see the Salesforce.com SOSL documentation.

Undelete

Restores deleted records from the Recycle Bin.

Update

Updates one or more records for existing business objects, such as accounts or contacts.

Usage notes and limitations:

- The Salesforce.com Partner SOAP API connector shows all fields of the selected sObject in Designer. However, at design time you must only select the fields that can be updated.
- The **Update** operation cannot update records for multiple object types.
- You must not use external ID fields as a foreign key to update a record and relate it to another existing record in a single step.

Upsert

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

Query

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

- `where`: This parameter can be used to contain a `where` clause.
- `limit`: This parameter can be used to contain the number of records to limit for the query.

The value can either be given at run time or you can specify a default value in the Parameters tab. The value given at run time always take precedence over the default value. However, if the existing default value is of type "fixed default", the overwrite will fail.

QueryAll

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

- `where`: This parameter can be used to contain a `where` clause.
- `limit`: This parameter can be used to contain the number of records to limit for the query.

The value can either be given at run time or you can specify a default value in the Parameters tab. The value given at run time always take precedence over the default value. However, if the existing default value is of type "fixed default", the overwrite will fail.

QueryMore

Retrieves the next batch of sObjects from a query.

Usage notes and limitations:

- **query**, **queryAll**, and **queryMore** do not support nested queries.

Utility Operations

Get Server Timestamp

Retrieves the server timestamp.

Get User Info

Returns standard information for the current user.

Reset Password

Changes the password to a system-generated value temporarily.

Set Password

Sets a specified value as your password.

7 CloudStreams Salesforce.com Bulk API Connector

- Overview 62
- REST Cloud Connections 62
- REST Resources 62

Overview

The CloudStreams Salesforce.com Bulk API 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.com Bulk API connector.

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

REST Cloud Connections

When starting a REST cloud connection, the Salesforce.com Bulk API connector invokes the Salesforce.com *login* operation. The *login* operation uses the values you specify when configuring a REST cloud connection in Integration Server Administrator. When you enable a connection, the CloudStreams server creates the login REST request envelope using the input mapping information. The CloudStreams server sends the login REST envelope to the Salesforce Server endpoint URL that is specified for the connection. The login response is parsed by applying the output mapping that extracts information, such as the server URL and session ID. The server URL and session ID are persisted in the CloudStreams REST cloud connection. The cloud connection is stored in the connection pool.

When stopping a REST cloud connection, the Salesforce.com Bulk API connector invokes the Salesforce.com *logout* operation. When disabling the cloud connection, the CloudStreams server creates the logout REST request envelope by using the input mapping information. The CloudStreams server sends the logout REST envelope to the Salesforce Server URL that is stored in the cloud connection during the login operation. After the logout operation is completed, the associated CloudStreams cloud connection is removed from the connection pool and the entire pool is destroyed.

REST Resources

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 what 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.

Input and Output Signature

The CloudStreams Bulk API connector contains an IS 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.com Bulk API connector. The IS document is used for all job-related resources, for example **CreateJob**.

The input values to the fields in the service signature map directly to the input values required by the Salesforce.com bulk APIs. 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.

Error and Fault Handling

For most of the REST resources, the output signature refers to an IS 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.com Bulk API connector returns a fault document with details about the error response. Also, the Salesforce.com Bulk API connector does not throw a ServiceException if a processing failure occurs. The Salesforce.com Bulk API 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.

DateTime Handling

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

Usage Notes

The CloudStreams server uses the Salesforce.com Bulk API connector XML schemas to locate the available resources and their respective IS document types, for example `wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_JobInfo`.

The Salesforce.com Bulk API connector uses the XML schemas, for example BulkREST.xsd, to generate IS documents that have all the required resource definitions for a given resource. The IS 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" on page 48.

Supported Resources

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

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.

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`.

Job

Aborts or closes an existing job.

Resource Definitions

`wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_JobInfo`

JobDetails

Retrieves the details of an existing job.

Resource Definitions

`wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_JobInfo`

BatchInfoList

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

Resource Definitions

`wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_BatchInfoList`

BatchStatus

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

Resource Definitions

`wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_BatchInfo`

BatchResult

Retrieves the Batch Result List.

Resource Definitions

The service signature is mapped to the `wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_BatchResult` IS document. This IS 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.

ResultList

Retrieves the results of a query batch.

Resource Definitions

The service signature is mapped to the `wmSalesforceBulkConnector_vXX.doctypes:docTypeRef_tns_ResultList` IS 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.

QueryResult

Retrieves the result of a query batch.

Resource Definitions

None.

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

GetBatchRequest

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

Resource Definitions

None.

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