webMethods Salesforce.com Adapter Installation and User's Guide

Version 7.1

April 2010

webMethods

This document applies to webMethods Salesforce.com Adapter Version 7.1 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: ADAPTER-SALESFORCE.COM-IUG-71-20100430

Table of Contents

About This Guide

This guide describes how to configure and use the webMethods Salesforce.com Adapter. 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:

- The Salesforce objects that you want to manage
- Salesforce.com workflow and workflow configurations
- Terminology and basic operations of your operating system
- Be familiar with the setup and operation of the webMethods Integration Server.
- Have a general idea about how to perform basic tasks with webMethods Developer or webMethods Designer.

Note: Procedures for creating flow services, adapter notifications, and adapter services are similar in Developer and Designer.

Document Titles

Some Software AG document titles have changed during product releases. The following table will help you locate the correct document for a release on the Software AG Documentation Web site or the Empower Product Support Web site.

Documentation	Title
Developer user's guide	■ For Developer 8.0 SP1 and later, use *Developing Integration Solutions: webMethods Developer User's Guide*. ■ For Developer 8.0 and earlier, use *webMethods Developer User's Guide*.
installation guide	■ For webMethods product suite 8.0 SP1 and later, use *Software AG Installation Guide*. ■ For webMethods product suite 8.0 and earlier, use *webMethods Installation Guide.*
Integration Server administration guide	■ For Integration Server 8.0 SP1 and later, use *Administering webMethods Integration Server*. ■ For Integration Server 8.0 and earlier, use *webMethods Integration Server Administrator's Guide*.

Documentation	Title
logging guide	■ For Integration Server 8.0 SP1 and later, use *webMethods Audit Logging Guide*. ■ For Integration Server 8.0 and earlier, use *webMethods Logging Guide.*
Optimize documentation	■ For Optimize for Infrastructure 8.0 SP1 and later, use *Administering webMethods Optimize* or *Optimizing BPM and System Resources with BAM: webMethods Optimize User's Guide*. ■ For Optimize for Infrastructure 8.0 and earlier, use *webMethods Optimize Administrator's Guide* or *webMethods Optimize User's Guide*.

Document Conventions

Convention	Description
Bold	Identifies elements on a user interface.
Narrow font	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.
[ ]	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. Depending on the release of the webMethods product suite, the location of the downloaded documentation will be as shown in the table below.

For webMethods...	The documentation is downloaded to...
6.x	The installation directory of each product.
7.x	A central directory named _documentation in the main installation directory (webMethods by default).
8.x	A central directory named _documentation in the main installation directory (Software AG by default).

Online Information

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

Note: The Empower Product Support Web site and the Software AG Documentation Web site replace Software AG ServLine24 and webMethods Advantage.

If you want to...	Go to...
Access the latest version of product documentation.	Software AG Documentation Web site 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. ■ Learn about critical alerts. See the Products area to: ■ Download products. ■ Get information about product availability. ■ Access older versions of product documentation. ■ Submit feature/enhancement requests.	Empower Product Support Web site https://empower.softwareag.com

If you want to...	Go to...
■ 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 Web sites 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/webmethods

1 Overview of the webMethods Salesforce.com Adapter

About the Adapter

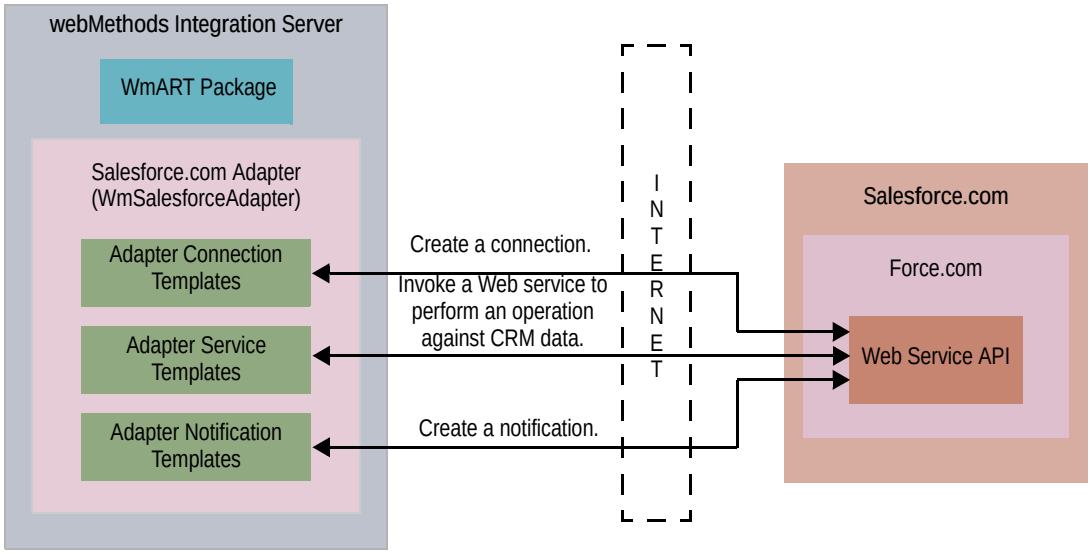
The webMethods Salesforce.com Adapter is an add-on to webMethods that enables you to interact with Salesforce.com. Salesforce.com is a Customer Relationship Management (CRM) solution that is Software as a Service (SaaS), on-demand, and Web-based.

Salesforce.com provides Force.com as a computing platform, from which it exposes access to its CRM data via a Web services API. Salesforce.com publishes a WSDL file to describe the Web services. The Salesforce.com Adapter uses the API to interact with Salesforce.com. Using adapter service templates, you can create adapter services that use the API to perform operations against the Salesforce.com CRM data. Your webMethods Integration Server clients can create and run services that use these adapter services to create, query, retrieve, update, and delete CRM data. For example, you can use the Salesforce.com Adapter to synchronize non-Salesforce.com CRM data with CRM data in Salesforce.com.

Architectural and Components

The Salesforce.com Adapter provides a set of user interfaces, services, and templates that enable you to create integrations with Salesforce.com. The adapter is provided as a single package that must be installed on Integration Server. For detailed installation instructions and software requirements, see Chapter 2, "webMethods Salesforce.com Adapter Installation".

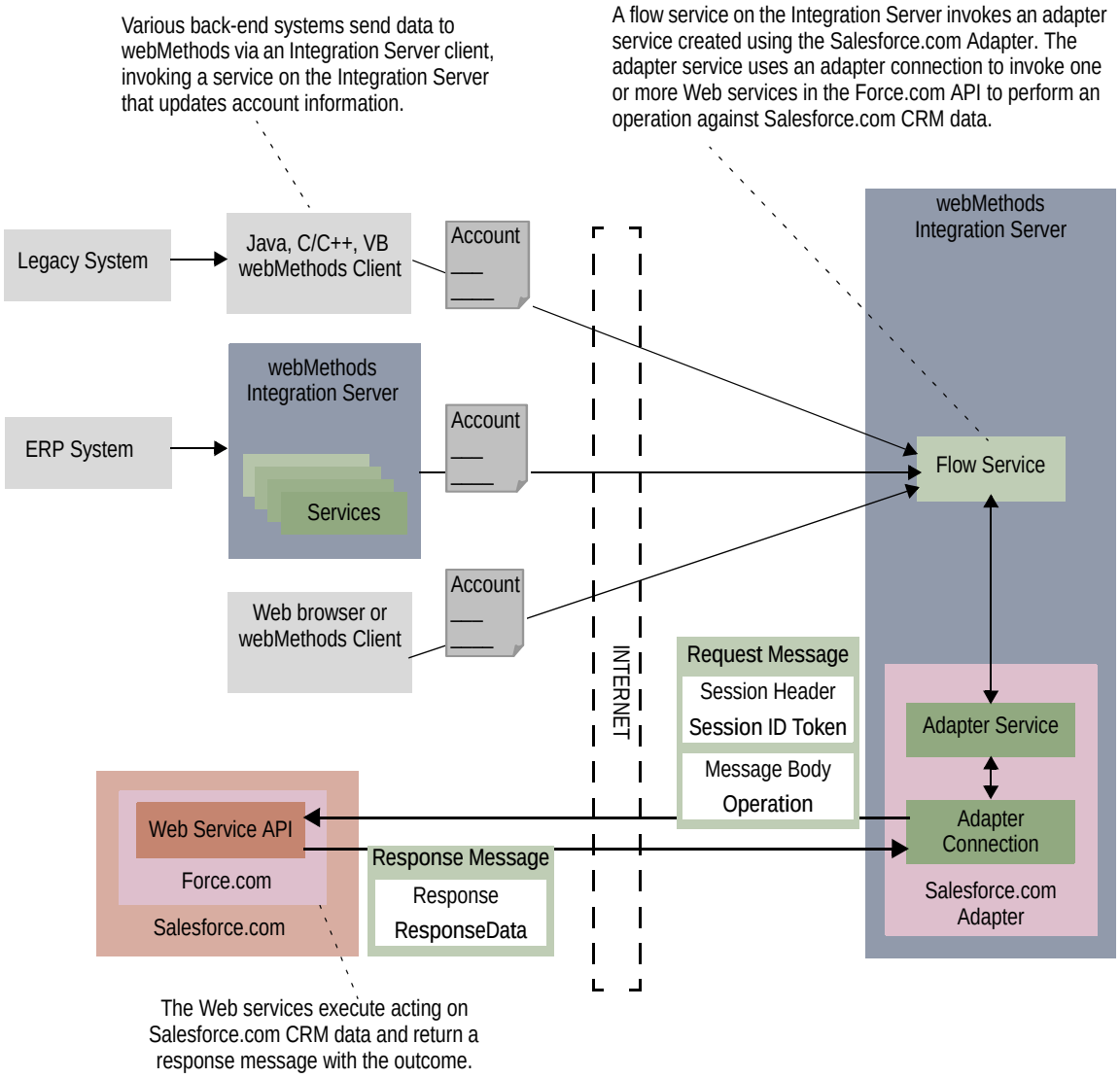
The following diagram shows at a high level how the adapter components connect to Salesforce.com:



- webMethods Integration Server. The Salesforce.com Adapter is installed and runs on the Integration Server.

- WmART Package. The WmART package provides a common framework for webMethods version 6 and later adapters to use the Integration Server's functionality, making the Integration Server the run-time environment for the Salesforce.com Adapter. The WmART package is installed with the Integration Server and provides logging, transaction management, and error handling for the adapter and its connections and services.
- Salesforce.com Adapter. The Salesforce.com Adapter is delivered as a single package called WmSalesforceAdapter. The adapter installation includes templates from which all adapter connections, adapter services, and adapter connections can be created. The adapter provides:
  - Integration Server Administrator user interfaces that enable you to configure and manage adapter connections
  - webMethods Developer and webMethods Designer user interfaces that enable you to configure and manage adapter services and notifications
  - Java stubs generated from the Partner WSDL version 13 used to channel service requests from the adapter to Salesforce.com
  - Apache Axis 1.4 libraries that the adapter uses when it is installed on Integration Server 6.5 SP3
- Adapter Connection Templates. An adapter connection enables the Integration Server to connect to Salesforce.com at run time. You must configure an adapter connection before you can create adapter services or notifications. The Salesforce.com Adapter provides templates for adapter connections in the Integration Server Administrator. For a detailed description of adapter connections and usage information, see "Adapter Connections" on page 15.
- Adapter Service Templates. An adapter service enables the Integration Server to execute Force.com Web services to interact with Salesforce.com. For example, an adapter service could update contact records within the Contact Salesforce object. The adapter provides adapter service templates in Developer or Designer. For more information about adapter service templates, see "Adapter Services" on page 17.
- Adapter Notification Templates. The Salesforce.com Adapter uses polling notifications, which poll Salesforce.com for data changes in the organization's data using the supported Data replication APIs, getUpdated() and getDeleted(). For a detailed description of adapter notifications, see"Adapter Notifications" on page 20.
- Salesforce.com. Salesforce.com Adapter uses the Force.com Web services API to access Salesforce.com CRM data. Salesforce.com publishes a Partner WSDL for the Web services describing the various supported APIs that can be invoked to access the CRM data. For information about the Salesforce.com components that the adapter requires, see "Requirements" on page 28.

The following diagram illustrates the use of the Salesforce.com Adapter and the Integration Server in a typical business process integration.



Adapter Package Management

The Salesforce.com Adapter is provided in the WmSalesforceAdapter package. You manage this package like any package on the Integration Server. Follow these steps to set up and effectively managing your packages on the Integration Server:

- Create user-defined packages for your connections, adapter services, and adapter notifications. See "Managing the Adapter Package" on page 34 for details.
- Understand how package dependencies work so you make the best decisions regarding how you manage your adapter services and adapter notifications. See "Package Dependency Requirements and Guidelines" on page 35 for details.

- Control which development groups have access to which adapter services and adapter notifications. See "Controlling Group Access" on page 38 for details.
- Understand how clustering, an advanced feature of the webMethods Integration Server, works to effectively manage your adapter services and adapter notifications. For more information, see "Using the Adapter in a Clustered Environment" on page 38.
- Enable and disable packages. For more information, see "Enabling and Disabling Packages" on page 36.
- Load, reload, and unload packages. For more information, see "Loading, Reloading, and Unloading Packages" on page 36.

Adapter Connections

You create one or more connections at design time to use in integrations. The number of connections you create depends on your integration needs.

Salesforce.com Adapter connections contain credentials that the Integration Server must supply to Salesforce.com when the adapter invokes Web services that act on the organization data. You configure connections using the Integration Server Administrator. You must have webMethods administrator privileges to access the administrative screens of the adapter.

When the Salesforce.com Adapter makes a login request, Salesforce.com sends a sessionId and serverURL as a response for making all the subsequent calls. The adapter connection maintains this sessionId and serverURL, which are required for making any other Web service calls to Salesforce.com. The sessionId is essential for Salesforce.com to ensure that the call is from an authenticated user.

Salesforce.com enforces a session timeout limit that invalidates the sessionId when a configured limit is reached. In order to keep the Salesforce.com connection valid, the adapter monitors the value of Session Timeout parameter and logs in again before the session times out. Maintaining the adapter connection ensures that the adapter service does not fail because of an INVALID_SESSION_ID error.

For instructions on configuring, viewing, editing, enabling, and disabling Salesforce.com Adapter connections, see Chapter 4, "Adapter Connections". For information about setting user privileges, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7. For a list of tasks that you must do before you can create your connections, see "Before Configuring or Managing Adapter Connections" on page 48.

Connection Pools

The 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 adapter connections.

A connection pool is a collection of connections with the same set of attributes. The Integration Server maintains connection pools in memory. Connection pools improve performance by enabling adapter services to reuse open connections instead of opening new connections. Depending on the type of edition subscribed by the organization, Salesforce.com limits the number of API requests or calls that any one organization may execute concurrently, or within any 24 hour period. Ensure that you set the Minimum Pool Size and Maximum Pool Size parameters of adapter connection appropriately. For information about these parameters, see "Configuring Adapter Connections" on page 48. For more information on API usage metering, see the *Force.com Web Services API Developer's Guide*.

Run-Time Behavior of Connection Pools

When you enable a connection, the Integration Server initializes the connection pool, creating the number of connection instances you specified in the Minimum Pool Size parameter of the connection. Whenever an adapter service needs a connection, the 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 Pool Increment Size), and adds them to the connection pool. If the pool is full (as specified in Maximum Pool Size), the requesting service will wait for the Integration Server to obtain a connection, up to the length of time specified in the Block Timeout parameter, until a connection becomes available. Periodically, the Integration Server inspects the pool and removes inactive connections that have exceeded the expiration period that you specified in Expire Timeout.

For information about configuring connections, see Chapter 4, "Adapter Connections".

Built-In Services for Connections

The Integration Server provides built-in services that enable you to programmatically control connections. You can use them to enable and disable a connection, and to return usage statistics, the current state (Enabled or Disabled), and error status for a connection. These services are located in the WmART package, in the pub.art.connection folder.

The built-in service setAdapterServiceNodeConnection enables you to change the connection associated with an adapter service. See "Changing the Connection Associated with an Adapter Service at Design Time" on page 17.

For details about the WmART services, see the *webMethods Integration Server Built-In Services Reference*.

Changing the Connection Associated with an Adapter Service at Design Time

If you want to assign a different connection to an adapter service after you configure the adapter service, use the setAdapterServiceNodeConnection, which is located in the WmART package's pub.art.service folder. This service allows you to change the specific connection associated with an adapter service at design time so that you do not need to create and maintain multiple adapter services.

Note: This built-in service can be run at design time only; do not use it within an Integration Server flow or Java service. You must run this service directly from Developer or Designer by selecting the service and running it.

For details about the setAdapterServiceNodeConnection service, see the *webMethods Integration Server Built-In Services Reference*.

Other built-in services enable you to control connections; for more information, see "Built-In Services for Connections" on page 16.

Changing the Connection Associated with an Adapter Service at Run Time

You can dynamically select the connection an adapter service uses to interact with the resource of the adapter. This feature enables one service to interact with multiple, similar back-end resources.

For example, you can configure an adapter service to use a default connection that interacts with the production Salesforce.com data. However, at run time, you can override the default connection and use another connection to interact with the Force.com sandbox for testing.

Alternatively, at run time, you can override the Salesforce.com user credentials configured for that connection by specifying an alternative user name and password to connect to Salesforce.com.

For more information about overriding a service's default connection at run time, see "Dynamically Changing a Service's Connection at Run Time" on page 53.

Adapter Services

To use the Salesforce.com Adapter, you need to configure adapter services. Adapter services allow you to connect to the adapter's resource (that is, Salesforce.com) and initiate an operation on the resource from the Integration Server.

You call adapter services from flow or Java services to interact with Salesforce.com. The Integration Server uses adapter connections that you defined earlier to execute the adapter services. The adapter services invoke Web services in the Force.com Web services API.

Adapter services are based on templates provided with the adapter. Each template represents a specific operation you can perform against Salesforce.com CRM data, such as retrieving field values from records in a Salesforce object.

An adapter service template contains all the code necessary for interacting with Salesforce.com, but without the data specifications. You provide these specifications when you create a new adapter service. To do so, you assign an adapter connection to the service using the webMethods Developer or webMethods Designer. Then select the adapter service template and supply the required data specifications.

For more information about working with services, see the *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.

Adapter Service Templates

The Salesforce.com Adapter provides the following adapter service templates:

Adapter Service Template	Description
Create Operation	Inserts new records into a Salesforce.com object. For example, you might insert new contacts in the Contacts Salesforce object. For more details, see "Configuring Create Operation Adapter Services" on page 60.
Delete Operation	Deletes records from the data of the organization. For example, you might delete one or more contacts from the Contacts Salesforce object. For more details, see "Configuring Delete Operation Adapter Services" on page 62.
Query Operation	Executes a query against a Salesforce object to find matching records and return field values from those records. For example, you might query the Contacts Salesforce object to find contacts that have a certain "Title" and return the values in the FirstName and LastName fields for those matching contacts. For more details, see "Configuring Query Operation Adapter Services" on page 64.
Retrieve Operation	Retrieves records of a Salesforce object and returns field values from those records. You specify the IDs of the records to retrieve. For more details, see "Configuring Retrieve Operation Adapter Services" on page 67.

Adapter Service Template	Description
Update Operation	Updates records of a Salesforce object. For example, you might update "Address Information" for a contact. For more details, see "Configuring Update Operation Adapter Services" on page 70.
Upsert Operation	Either inserts records into or updates records of a Salesforce object. You specify how to identify the record on which to act. If the record does not already exist, it is inserted. If the record exists, it is updated. For example, you might identify a contact by its external ID and provide information for the contact. If the external ID does not identify an existing contact, the contact is added; otherwise, the contact with the specified external ID is updated with the supplied information. For more details, see "Configuring Upsert Operation Adapter Services" on page 72.
Utility Operation	Invokes one of the following Salesforce.com utilities: ■ getServerTimestamp. Retrieves the current Salesforce.com system timestamp. ■ getUserInfo. Obtains personal information about the currently logged-in user, that is, the user account used for the adapter connection. ■ resetPassword. Changes a user's password to a temporary, system-generated value. ■ sendEmail. Sends a single e-mail message. ■ sendMassEmail. Sends multiple e-mail messages. ■ setPassword. Changes a user's password to a specified value. For more details, see "Configuring Utility Operation Adapter Services" on page 75.

Note: Making a call to the resetPassword or setPassword utilities changes the credentials of the user, and will cause any new login requests to fail. Also, any new reset or set password operation resets the security token. You should update the password field in the connection configuration with a new password and security token. This security token is not required while testing on the Salesforce.com sandbox account.

Adapter Notifications

The Salesforce.com Adapter supports polling notifications that monitor a specified sObject for changes, such as an insert, update, or delete operation, so that the appropriate Java or flow services can use the data for operations such as sending an invoice, or publishing it to the Integration Server.

A polling notification periodically checks the resource at specified intervals for the occurrence of events, and publishes a document each time an event occurs in the resource. For example, your application might need to be notified when data is added, updated, and deleted from the resource. Integration Server invokes the notification periodically, based on the polling interval that you specify when you configure the notification.

Use the Developer or Designer to configure, edit, and delete polling notifications. All notifications are configured from a notification template and require a configured connection.

For more information, see Chapter 6, "Adapter Notifications".

Choice of Publish Destinations

Beginning with Integration Server 8.0, you can choose the destination to which asynchronous notifications should publish messages. Specifically, you can choose whether the asynchronous notification templates use JMS APIs to publish messages to the Integration Server JMS messaging provider or Broker APIs to publish notification messages to webMethods Broker.

Note: To use the JMS protocol with asynchronous notifications, you must first configure a JMS connection alias on Integration Server. For more information, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7.

For steps for selecting a publish destination for asynchronous notification messages, see Chapter 6, "Adapter Notifications".

Adapter Notification Templates

The Salesforce.com Adapter provides the following adapter notification templates:

Notification Type	Notification Template	Description
Upsert Notification	UpsertNotification	Wraps the Force.com Web service API-getUpdated() call. Publishes notification of insert and update operations on a Salesforce object.
Delete Notification	DeleteNotification	Wraps the Force.com Web service API-getDeleted() call.

These polling notifications can be configured only for those Salesforce objects that can be replicable (that is, the replicateable field for the Object is true). For information about replicating objects, see the *Force.com Web Services API Developer's Guide*.

Notification Types

The Salesforce.com Adapter provides two types of notifications: Upsert and Delete.

Using Upsert and Delete Notifications

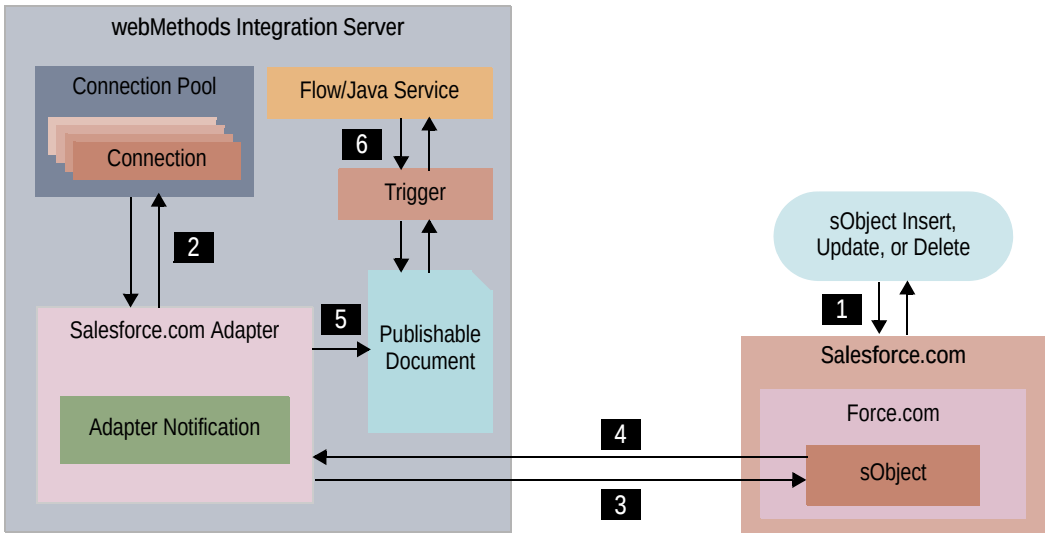
The following table lists the tasks required to use these types of notification:

	Task	Use this tool...
1	Create an adapter connection. See "Configuring Adapter Connections" on page 48 for details.	Integration Server Administrator
2	Configure the notification and specify the following: ■ Adapter connection. ■ Salesforce object. ■ Publishable document to contain the data from Salesforce.com. There is a single publishable document used for all events associated with the notification. For more details about Integration Server publishable documents, see the *Publish-Subscribe Developer's Guide*. ■ Output data fields contained in the publishable document. See "Configuring Adapter Notifications" on page 85 for instructions for configuring notifications.	Developer or Designer
3	If you plan to use an Integration Server flow or Java service, design it to react to the data changes contained in the notification's publishable document. Create the Integration Server trigger to use the notification's publishable document. For details, see the *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.	Developer or Designer

Task		Use this tool...
4	Schedule and enable the adapter notification. When you enable the notification, the Integration Server Scheduler invokes the notification and continues to do so periodically, based on the polling schedule parameters you specify. See "Managing Polling Notifications" on page 88 for instruction to schedule and enable notifications.	Integration Server Administrator
5	Manage the notification. See Chapter 6, "Adapter Notifications" for details.	Developer or Designer and Integration Server Administrator

Upsert and Delete Notifications Processing

The following diagram and steps illustrate what happens when these types of notifications are invoked. The Integration Server continues to invoke the notification periodically, as defined when you configured the polling schedule parameters for the notification.



Step	Description
1	Upsert Notifications and Delete Notifications monitor the organization data modified in Salesforce.com. You specified the sObject to be monitored when you first configured the adapter notification.
2	The notification gets a connection from the connection pool of the service. Adapter connections contain the connection information for Salesforce.com.

Step	Description
3	The notification uses the Apache Axis API to connect to Salesforce.com. You created and enabled the adapter connection earlier using the Integration Server Administrator.
4	The notification retrieves sObject data from Salesforce.com.
5	The notification creates the publishable document, which contains data from Salesforce.com. The notification publishes the publishable document. For more details about Integration Server publishable documents, see the *Publish-Subscribe Developer's Guide*.
6	Using an Integration Server trigger, you configured to use the notification's publishable document, a flow or Java service on the Integration Server is invoked to react to the data changes contained in the publishable document.

Support for Receiving Outbound Messages from Salesforce.com

webMethods Integration Server supports the outbound messaging feature of Salesforce.com. Using this feature, you can configure Salesforce.com to send message notifications when changes are made to Salesforce objects. In certain circumstances, this messaging feature provides advantages over Adapter notifications.

In Salesforce.com, you can configure a workflow rule that defines what changes should trigger a notification, and indicates that you want Salesforce.com to send an outbound message when those changes occur. For instructions on how to configure Salesforce.com to send outbound message notifications to your Integration Server, see Chapter 7, "Support for Salesforce.com Outbound Messaging".

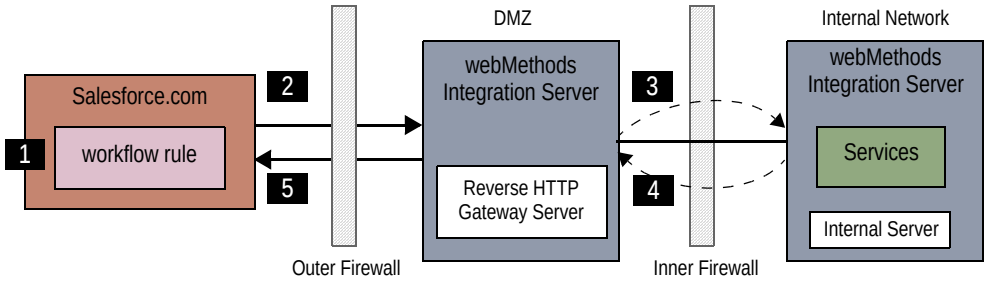
The outbound notification messages from Salesforce.com are useful when you want to specify the criteria for the notification being sent to Integration Server. Salesforce.com continuously monitors the designated sObject and sends an outbound message to Integration Server when the specified field values of the monitored sObject are changed, and the defined workflow rule is executed. In contrast, Adapter notifications retrieve only the IDs of the changed records by polling at configured intervals.

Note that in some cases, adapter notifications offer advantages over Salesforce.com messages. One advantage of using Adapter notifications is that by specifying the start date for polling notification, you have access to data changes made in the past. It is not possible to get the field values changed in the past using the outbound messages from Salesforce.com. Each outbound message contains information about the data changed in a single record, whereas adapter notifications can retrieve information from multiple records.

To ensure security when receiving notification messages from Salesforce.com, place your Integration Server behind an internal firewall and place a Reverse HTTP Gateway in the DMZ to allow it to accept communications and process requests from Salesforce.com.

In a Reverse HTTP Gateway configuration, your Integration Server, which is referred to as the *Internal Server*, remains behind your inner firewall where external clients cannot access it. You place another Integration Server in your DMZ to act as a *Reverse HTTP Gateway Server*. The Reverse HTTP Gateway Server acts as an intermediary between the Internet and your Internal Server.

The following diagram illustrates the use of outbound message notifications when using a Reverse HTTP Gateway Server configuration.



Step	Description
1	In Salesforce.com, configure a workflow rule to monitor the changes in a Salesforce object for which you want notification. Configure the workflow rule to send an outbound message when the change occurs. When defining the outbound message, specify the gateway external port (or the proxy port in case of Integration Server 6.5) as the endpoint URL.
2	When Salesforce.com detects the change, it sends the outbound message to the endpoint URL configured in the workflow rule. Because the Reverse HTTP Gateway Server is configured, the outbound message first goes to the Reverse HTTP Gateway Server. The Reverse HTTP Gateway Server passes the outbound message to the Internal Server.
3	The listener service consumes the outbound message. The business logic in the listener service performs processing to handle the outbound notification message.
4	In addition to performing processing to handle the notification, the listener also sends a SOAP response message back to Salesforce.com. The SOAP response message first goes to the Reverse HTTP Gateway Server.
5	The Reverse HTTP Gateway Server passes the SOAP response message to Salesforce.com.

For more information about using a Reverse HTTP Gateway Server, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7. For more information on setting up Reverse HTTP Gateway Server to receive outbound messages from Salesforce.com, see Chapter 7, "Support for Salesforce.com Outbound Messaging".

Adapter Notification vs. Outbound Message Notification from Salesforce.com

This section lists the differences between the adapter notification and the outbound message notification feature of Salesforce.com.

Functionality	Adapter Notification	Outbound Message Notification from Salesforce.com
Immediate notification	Not available	Available
Security	Available	Available only if you are using a Reverse HTTP Gateway Server.
Access to past data changes	Available	Not available
Bulk data retrieval	Available	Not available
Conditional notification	Not available	Available

webMethods Infrastructure Data Collector Support

webMethods Infrastructure Data Collector monitors the system and operational data associated with webMethods run-time components such as Integration Servers, Broker Servers, Brokers, and adapters, and reports the status of these components on Optimize for Infrastructure or other external tools. When you start monitoring an Integration Server, Infrastructure Data Collector automatically starts monitoring all ART-based adapters that are installed on the Integration Server.

For information about monitored key performance indicators (KPIs) collected for the monitored adapter components, see the appropriate Optimize documentation for your release in "Document Titles" on page 7.

Viewing the Adapter's Update Level

You can view the list of updates that have been applied to the adapter. The list of updates appears in the Updates field on the adapter's About page in the Integration Server Administrator.

Controlling Pagination

When using the adapter on Integration Server 8.0 and later, you can control the number of items that are displayed on the adapter Connections screen and Notifications screen. By default, 10 items are displayed per page. Click Next and Previous to move through the pages, or click a page number to go directly to a page.

To change the number of items displayed per page, set the watt.art.page.size property and specify a different number of items.

To set the number of items per page

1 From Integration Server Administrator, click Settings > Extended.

2 Click Edit Extended Settings. In the Extended Settings editor, add or update the watt.art.page.size property to specify the preferred number of items to display per page. For example, to display 50 items per page, specify:

```
watt.art.page.size=50
```

3 Click Save Changes. The property appears in the Extended Settings list.

For more information about working with extended configuration settings, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7.

2 webMethods Salesforce.com Adapter Installation

Overview

This chapter, in conjunction with the *Software AG Installation Guide*, explains how to install and uninstall webMethods Salesforce.com Adapter 7.1. If you are installing the adapter with webMethods Integration Server (prerequisite) or other webMethods products, this *webMethods Salesforce.com Adapter Installation and User's Guide* refers to the *Software AG Installation Guide*, but you should use the correct installation guide as follows:

- If you are installing the adapter with webMethods 8 products, refer to the *Software AG Installation Guide* for instructions.
- If you are installing the adapter with previous versions of webMethods products, refer to the *webMethods Installation Guide* for instructions.

Requirements

For a list of the operating systems and webMethods products supported by the adapter, see the *webMethods Adapters System Requirements*, available in the webMethods area of the Software AG Documentation Web page.

Salesforce.com Adapter 7.1 has no hardware requirements beyond those of its host Integration Server.

The following table lists the Salesforce.com components that are required by the Salesforce.com Adapter.

Component	Comment
Salesforce.com Partner WSDL 13	When defining an adapter connection, be sure to specify the EndPoint URL to point to version 13.0. The default EndPoint used by Salesforce.com Adapter is: https://www.salesforce.com/services/Soap/u/13.0 When using an alternate EndPoint, like Salesforce.com Sandbox or test environment, be sure to use version 13.0. Specifying a different version might give unexpected results.

Component	Comment
Salesforce.com API-enabled user account	When defining an adapter connection, you must specify the credentials of a Salesforce.com user account that the adapter uses when interacting with Salesforce.com using the Partner WSDL API. The Salesforce.com Adapter requires that the API feature must be enabled by the user organization. This feature is enabled by default for Unlimited, Enterprise, and Developer Editions. Some Professional Edition organizations might also have the API enabled. If the API Enabled feature is not available, contact Salesforce.com and ensure that it is enabled before using this adapter.
Apache Axis 1.4	When running the adapter on Integration Server 7.1.2 and later, the adapter uses the common Integration Server libraries. When running the adapter on Integration Server 6.5 SP3, the Apache Axis 1.4 libraries are bundled with the adapter package (WmSalesforceAdapter).

Installing Salesforce.com Adapter 7.1

The instructions in this section are meant to be used with the more complete instructions in the *Software AG Installation Guide*. The instructions explain how to use the Software AG Installer wizard.

Note: If you are installing the adapter in a clustered environment, you must install it on each Integration Server in the cluster, and each installation must be identical. For more information about working with the adapter in a clustered environment, see "Using the Adapter in a Clustered Environment" on page 38.

To install Salesforce.com Adapter 7.1

1 Download the Installer from the Empower Product Support Web site

2 If you are installing the adapter on an existing Integration Server, shut down the Integration Server.

3 Start the Installer wizard.

- Choose the webMethods release that includes the Integration Server on which to install the adapter. For example, if you want to install the adapter on Integration Server 8.0, choose the 8.0 release.

- If you are installing on an existing Integration Server, specify the webMethods installation directory that contains the host Integration Server. If you are installing both the host Integration Server and the adapter, specify the installation directory to use. The Installer will install the adapter in the *Integration Server_directory*\packages directory.

4 Install the adapter:

- On Integration Server 7.1.2 and later, in the product selection list, select **Adapters > webMethods Salesforce.com Adapter 7.1**. Documentation for the adapter is available by selecting **Documentation > Adapter Readmes and Documentation** from the product selection list. Note that this will install the documentation for all adapters. The Installer will install the documentation files in the *installation_directory*_documentation directory.
- On Integration Server 6.5 SP3, in the product selection list, select **Adapters > webMethods Salesforce.com Adapter 7.1 > Program Files**. You can also select **Documentation**. The Installer will install the documentation in the *Integration Server_directory*\packages\WmSalesforceAdapter\doc directory.

Note: You can download the adapter documentation at a later time from the Software AG Documentation Web site (http://documentation.softwareag.com/).

5 After the installation completes, close the Installer.

6 If you want to use the compression feature (as explained in "Use Compression" on page 51), download the commons-httpclient.jar and commons-codec.jar files from the Apache download center. Copy the files to the appropriate directory below:

Integration Server	Copy the files to this directory...
8.0, 7.1.2	*Integration Server_directory*\lib\jars
6.5 SP3	*Integration Server_directory*\packages\WmSalesforceAdapter\code\jars

7 If you are using Developer 6.5 SP4, download the latest fix from the Empower Product Support Web site and apply it.

8 Start the host Integration Server.

9 Make sure the WmTomcat package is enabled. For instructions, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7.

Uninstalling Salesforce.com Adapter 7.1

The instructions in this section are meant to be used with the uninstallation instructions in the *Software AG Installation Guide*.

To uninstall Salesforce.com Adapter 7.1

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 the Software AG Uninstaller, selecting the webMethods installation directory that contains the host Integration Server. In the product selection list, select Adapters > webMethods Salesforce.com Adapter 7.1. You can also choose to uninstall documentation.
3. After the Uninstaller completes, restart the host Integration Server.
4. The Uninstaller removes all Salesforce.com Adapter 7.1-related files that were installed. However, it does not delete files created after you installed the adapter (for example, user-created or configuration files), nor does it delete the adapter directory structure. You can go to the *Integration Server_directory*\packages directory and delete the WmSalesforceAdapter directory.

3 Adapter Package Management

Overview

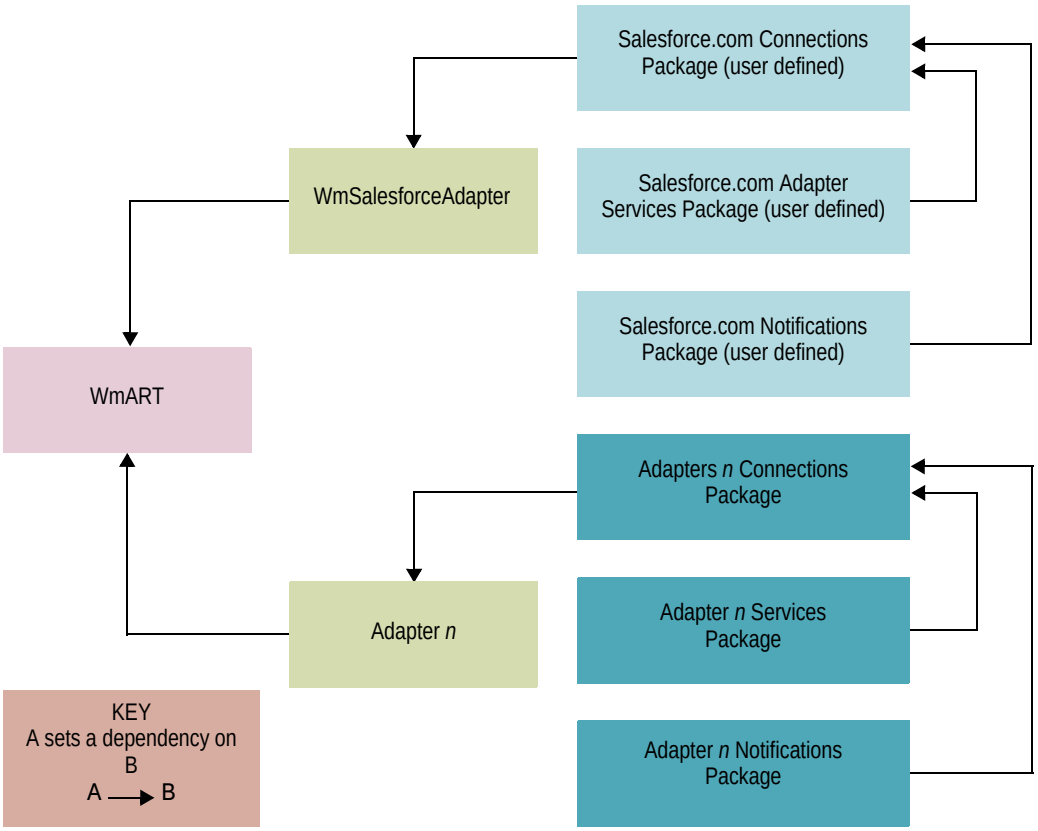
The following sections describe how to set up and manage your webMethods Salesforce.com Adapter packages, set up Access Control Lists (ACL), and use the adapter in a clustered environment.

Managing the Adapter Package

The Salesforce.com Adapter is provided in the WmSalesforceAdapter package. Manage the WmSalesforceAdapter package the same way you would manage any Integration Server package.

Define connections, adapter services, and adapter notifications in user-defined packages rather than in the WmSalesforceAdapter package. Doing so will allow you to manage the package more easily.

As you create user-defined packages in which to store connections, adapter services, and adapter notifications, use the package management functionality provided in the Developer and set the user-defined packages to have a dependency on the WmSalesforceAdapter package. That way, when the WmSalesforceAdapter package loads or reloads, the user-defined packages load automatically. See the following diagram:



Package management tasks include:

- Setting package dependencies (see "Package Dependency Requirements and Guidelines" on page 35).
- "Enabling and Disabling Packages" on page 36.
- "Controlling Group Access" on page 38.

Package Dependency Requirements and Guidelines

This section contains a list of dependency requirements and guidelines for user-defined packages. For instructions for setting package dependencies, see the appropriate Developer user's guide for your release in "Document Titles" on page 7.

- By default, the WmSalesforceAdapter package has a dependency on the WmART package. Do not change this dependency.
- A user-defined package must have a dependency on its associated adapter package, WmSalesforceAdapter. (The WmSalesforceAdapter package has a dependency on the WmART package.)
- Package dependencies ensure that at startup the Integration Server automatically loads or reloads all packages in the proper order: the WmART package first, the adapter package next, and the user-defined packages last. The WmART package is automatically installed when you install the Integration Server. You should not need to manually reload the WmART package.
- If the connections and adapter services of an adapter are defined in different packages, then:
  - A package that contains the connections must have a dependency on the adapter package, WmSalesforceAdapter.
  - Packages that contain adapter services must have a dependency on their associated connection package.
- Keep connections for different adapters in separate packages so that you do not create interdependencies between adapters. If a package contains connections for two different adapters, and you reload one of the adapter packages, the connections for both adapters will reload automatically.
- The Integration Server will not allow you to enable a package if it has a dependency on another package that is disabled. That is, before you can enable your package, you must enable all packages on which your package depends. For information about enabling packages, see "Enabling and Disabling Packages" on page 36.
- The Integration Server *will* allow you to disable a package even if another package that is enabled has a dependency on it. Therefore, you must manually disable any user-defined packages that have a dependency on the adapter package before you disable the adapter package.
- You can name connections, adapter services, and notifications the same name provided that they are in different folders and packages.

Enabling and Disabling Packages

All packages are automatically enabled by default. When you want to temporarily prohibit access to the elements in a package, disable the package. When you disable a package, the server unloads all of its elements from memory. Disabling a package prevents the Integration Server from loading that package at startup. A disabled package will remain disabled until you explicitly enable it using the Integration Server Administrator.

To enable a package

1 In the Integration Server Administrator select Packages > Management.

2 Click No in the Enabled column. The server displays a  and Yes in the Enabled column.

Note: Enabling an adapter package will *not* cause its associated user-defined packages to be reloaded. For information about reloading packages, see "Controlling Group Access" on page 38.

Important! Before you manually enable a user-defined package, you must first enable its associated adapter package (WmSalesforceAdapter). Similarly, if your adapter has multiple user-defined packages, and you want to disable some of them, disable the adapter package first. Otherwise, errors will be issued when you try to access the remaining enabled user-defined packages.

To disable a package

1 In the Integration Server Administrator select Packages > Management.

2 Click Yes in the Enabled column for the package that you want to disable. The server issues a prompt to verify that you want to disable the package. Click OK to enable the package. When the package is disabled, the server displays No in the Enabled column.

A disabled adapter will:

- Remain disabled until you explicitly enable it using the Integration Server Administrator.
- Not be listed in the Developer or Designer.

Loading, Reloading, and Unloading Packages

As described above in "Package Dependency Requirements and Guidelines" on page 35, if user-defined packages are properly configured with a dependency on the adapter package, at startup the Integration Server automatically loads or reloads all packages in the proper order: the WmART package first, the adapter package next, and the node packages last. You should not need to manually reload the WmART package.

Reloading Packages Manually

Reloading a user-defined package will *not* cause its associated adapter package to be reloaded. You can reload adapter packages and user-defined packages from either the Integration Server Administrator (by clicking the Reload icon on the Management window) or from Developer or Designer (by right-clicking the package and selecting the Reload Package option from the menu).

Unloading Packages

At shutdown, the Integration Server unloads packages in the reverse order in which it loaded them: it unloads the node packages first, the adapter package next, and the WmART package last (assuming the dependencies are correct).

Setting Package Dependencies

If a package requires services in another package, the other package must load before the dependent package can load. Set package dependencies to ensure packages are loaded in the correct order. For example, any packages you create for Salesforce.com Adapter services should identify the WmSalesforceAdapter package as a package dependency because they require services in the WmSalesforceAdapter to load first. Use the following guidelines:

- Set package dependencies from the adapter service package to the package containing the connection if you configure a connection in one package and the adapter services in another package. That is, the package that contains the connection should load before the adapter service package.

  When you set this package dependency, it ensures that if someone disables the connection package and then re-enables it, the adapter services will reload correctly.

- If both the connection and adapter services are in the same package, set this package to have a dependency on the WmSalesforceAdapter package.

- In general, packages containing connections should have a dependency set to the adapter package itself. That is, the adapter service package should depend on the adapter connection package, which should depend on the adapter package. Similarly, if the adapter services are in the same package as the connections, the only dependency that you need to set is between the adapter connection package and the adapter package.

For more information about setting package dependencies, see *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.

Controlling Group Access

To control which development group has access to which adapter services, use access control lists (ACLs). You can use ACLs to prevent one development group from inadvertently updating the work of another group, or to allow or deny access to services that are restricted to one group but not to others.

For general information about assigning and managing ACLs, see *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.

Using the Adapter in a Clustered Environment

What is webMethods Integration Server Clustering?

Clustering is an advanced feature of the webMethods product suite that substantially extends the reliability, availability, and scalability of webMethods Integration Server. Clustering accomplishes this by providing the infrastructure and tools to deploy multiple Integration Servers as if they were a single virtual server and to deliver applications that leverage that architecture.

With clustering, you get the following benefits:

- Reliability.Unlike a server farm (an independent set of servers), clustering provides the reliability required for mission-critical applications. Distributed applications must address network, hardware, and software errors that might produce duplicate (or failed) transactions. Clustering makes it possible to deliver "exactly once" execution as well as checkpoint/restart functionality for critical operations.
- Availability.Clustering provides virtually uninterrupted availability by deploying applications on multiple Integration Servers; in the worst case, a server failure produces degraded but not disrupted service.Without clustering—even with expensive Fault-Tolerant systems—a failure of the system (hardware, java runtime, or software) may result in unacceptable downtime.
- Scalability.Clustering provides horizontal scalability, which allows virtually limitless expansion of capacity by simply adding more machines of the same or similar capacity. Without clustering, only vertical scalability is possible. That is, increased capacity requirements can only be met by deploying on larger, more powerful machines, typically housing multiple CPUs. Salesforce.com enforces a usage metering for calls via its API, in order to balance transaction loads. This is done by limiting the number of API requests (or calls) that any one organization may execute concurrently or within any 24-hour period. These limits are based on the type of edition subscribed by an organization and are against the aggregate of all API calls, and not on a per user basis. When an organization exceeds the limit, the users may be temporarily blocked from making additional calls.

For more information about API usage metering by Salesforce.com, see the *Force.com Web Services API Developer's Guide*.

Load balancing is an optimizing feature you use with clustered Integration Servers. Load balancing controls how requests are distributed to the servers in the cluster. You must use a third-party load balancer to perform load balancing.

Note: Integration Server clustering redirects HTTP and HTTPS requests, but does not redirect SMTP requests.

For details on Integration Server clustering, see the *webMethods Integration Server Clustering Guide*.

Configuring the Adapter in a Clustered Environment

When you configure the Salesforce.com Adapter to create adapter services, you must:

- Ensure that each Integration Server in the cluster contains an identical set of packages (see "Replicating Packages to webMethods Integration Servers" on page 39).
- Enable multiple instances of the same polling notification in your cluster (see "Polling Notification Support in Integration Server Clusters" on page 41).

Replicating Packages to webMethods Integration Servers

Every webMethods Integration Server in the cluster should contain an identical set of packages that you define using the Salesforce.com Adapter; that is, you should replicate the Salesforce.com Adapter services, notifications, and the connections.

To ensure consistency, you create all packages on one server, and replicate them to the other servers. If you allow different servers to contain different services, you might not derive the full benefits of clustering. For example, if a client requests a service that resides in only one server, and that server is unavailable, the request cannot be successfully redirected to another server.

For information about replicating packages, see the chapter on managing packages in the appropriate Integration Server administration guide for your release in "Document Titles" on page 7.

Clustering Considerations and Requirements

Note: The following sections assume that you have already configured the webMethods Integration Server cluster. For details about webMethods clustering, see the *webMethods Integration Server Clustering Guide*.

The following considerations and requirements apply to the Salesforce.com Adapter in a clustered environment.

Requirements for Each Integration Server in a Cluster

The following table describes the requirements of each Integration Server in a given cluster:

All Integration Servers in a given cluster must have identical...	For Example...
Integration Server versions	One Integration Server in the cluster cannot be version 7.1.2 and another Integration Server in the cluster be version 6.5 SP3.
Adapter packages	All adapter packages on one Integration Server should be replicated to all other Integration Servers in the cluster.
Adapter versions	All Integration Servers in the cluster must have the same version of the adapter, with the same fixes (updates and service packs) applied.
Adapter connections	With Salesforce.com there is no physical connection. A Salesforce.com Adapter connection simply maintains the session binding details. If you plan to use connection pools in a clustered environment, see "Considerations When Configuring Connections with Connection Pooling Enabled" on page 41.
Adapter services	If you configure a specific adapter service, this same adapter service must appear on all servers in the cluster so that any Integration Server in the cluster can handle the request identically. If you allow different Integration Servers to contain different services, you might not derive the full benefits of clustering. For example, if a client requests a service that resides on only one server, and that server is unavailable, the request cannot be successfully redirected to another server.
Adapter notifications	If you configure a specific adapter notification, this same adapter notification must appear on all servers in the cluster. For more information, see "Polling Notification Support in Integration Server Clusters" on page 41.

See "Replicating Packages to webMethods Integration Servers" on page 39 for information about replicating adapter packages, connections, and adapter services across multiple Integration Servers in a cluster.

Considerations When Installing Salesforce.com Adapter Packages

For each Integration Server in the cluster, use the standard Salesforce.com Adapter installation procedures for each machine, as described Chapter 2, "webMethods Salesforce.com Adapter Installation".

Considerations When Configuring Connections with Connection Pooling Enabled

With Salesforce.com there is no physical connection. A Salesforce.com Adapter connection simply maintains the session binding details. However, there are restrictions imposed by Salesforce.com for concurrent API access. For information about API usage monitoring, see the *Force.com Web Services API Developer's Guide*.

For information about configuring connections for the Salesforce.com Adapter, see "Configuring Adapter Connections" on page 48.

For more general information about connection pools, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7.

Polling Notification Support in Integration Server Clusters

The Salesforce.com Adapter enables the coordinated execution of polling notifications within a webMethods Integration Server cluster. The Salesforce.com Adapter provides the ability to enable multiple instances of the same polling notification in your cluster, and to coordinate their schedules and execution. This provides enhanced quality of service by allowing configurations for automated failover between notifications and distributed processing of polling notifications.

You can configure a polling notification to run in either Standby or Distributed mode. You can also configure additional settings for clustered polling notifications.

Standby Mode and Distributed Mode

In Standby mode, a particular instance of a polling notification will execute the notification according to its configured schedule. When you start the cluster, the polling notification that executes the first scheduled run is considered to be the primary notification. This instance will continue to execute the scheduled run as long as it is enabled and fully functional. If at any time this notification becomes disabled, another notification in the cluster will assume control. The notification that assumes control is arbitrary. After a notification has control, it will continue to execute the schedule for as long as it is enabled and fully functional.

In Distributed mode, *any* instance of the polling notification can execute the currently scheduled run. The notification that executes the current scheduled run is arbitrary. If a notification does not complete executing within the amount of time specified in the Max Process Time field, the system considers that notification to be "dead." (For details about

Max Process Time, see "Notification-Specific Settings" on page 44.) Another (enabled) instance in the cluster will recognize this situation and will attempt to execute the scheduled run.

Settings

All settings that pertain to clustered polling notifications are ignored or disabled until you include the server in a cluster. All settings have default values. There are three levels of settings: server-wide, adapter-specific, and notification-specific. Additionally,

Server-Wide Settings

The server-wide settings are common to all webMethods 6.x and later adapters running on the Integration Server. The server uses the settings in an algorithm that determines whether a polling notification instance should be considered "dead." For more information, see the *webMethods Integration Server Clustering Guide*.

Server-wide Setting Name	Values and Description
watt.art.clusteredPollingNotification.keepAliveInterval	An integer value specifying the frequency (in milliseconds) at which a notification instance tells the cluster it is still alive.
watt.art.clusteredPollingNotification.keepAliveExpireTimeout	An integer value specifying the number of milliseconds that a keepAliveInterval setting can be late before it is assumed that an instance has failed. Default: The value of keepAliveInterval. Note: You should allow for "clock drift." For details, see "Clock Synchronization" on page 45.

Adapter-Specific Settings

The adapter-specific settings apply to all the polling notifications in your Salesforce.com Adapter.

The Salesforce.com Adapter generates a configuration file for your polling notification templates. This file, WmSalesforceAdapter\config\clusterProperties.cnf, is an XML file that contains a pair of settings (callbackScheme and runtimeModeLimit) for each polling notification template defined in your adapter. For example, if you created notifications using the notification templates UpsertNotification and DeleteNotification, you will see these settings for each template. For an example file, see "Example Salesforce.com Adapter clusterProperties.cnf File" on page 43.

The available values for these settings are as follows:

Adapter-specific Setting Name	Values and Description
callbackScheme	Specifies the run-time operations the adapter should handle for the notifications, such as enabling and disabling the notifications. ■ 0: No callback coordination. ■ 1: Default. Coordinates the enable and disable operations. ■ 2: Coordinates the startup and shutdown operations. ■ 3: Coordinates the enable, disable, startup, and shutdown operations. **Important!** This value must *always* be 1 for the Salesforce.com Adapter.
runtimeModeLimit	Specifies the scheduling mode for the notifications. The value you assign to this setting determines which modes that the adapter users may select in the Coordination Mode field on the Polling Notification Schedule page (see "Notification-Specific Settings" on page 44). ■ disable: Prevents the adapter user from selecting Standby or Distribute modes on the adapter's Polling Notification Schedule page. ■ standby: Default. Enables the adapter user to select either the Disable mode or the Standby mode. ■ distribute: Enables the adapter user to select either the Disable, Standby or Distribute mode. For more information, see "Standby Mode and Distributed Mode" on page 41.

Example Salesforce.com Adapter clusterProperties.cnf File

The following example clusterProperties.cnf file for the Salesforce.com Adapter shows the entries for all the possible templates associated with the Salesforce.com Adapter polling notifications. This example file enables all notifications to be configured with the Coordination Mode field set to distribute.

```
<?xml version="1.0"?>
<clusterProps>
  <pollingNotifications>
    <callbackScheme>1</callbackScheme>
    <runtimeModeLimit>distribute</runtimeModeLimit>
```

```
    <template
classNameName="com.wm.adapter.wmsalesforce.notification.DeleteNotification">
      <callbackScheme>1</callbackScheme>
      <runtimeModeLimit>distribute</runtimeModeLimit>
    </template>
    <template
classNameName="com.wm.adapter.wmsalesforce.notification.UpsertNotification">
      <callbackScheme>1</callbackScheme>
      <runtimeModeLimit>distribute</runtimeModeLimit>
    </template>
  </pollingNotifications>
  <listenerNotifications>
    <callbackScheme>1</callbackScheme>
  </listenerNotifications>
  <listeners>
    <runtimeModeLimit>distribute</runtimeModeLimit>
  </listeners>
</clusterProps>
```

Notification-Specific Settings

The notification-specific settings enable you to configure certain scheduling aspects of polling notifications on an individual basis.

Two new fields appear on the Polling Notification Schedule page: Coordination Mode and Max Process Time. These fields become editable when you add your Integration Server to a cluster.

- The Coordination Mode field controls the coordination of the notification schedules across the cluster. Depending on the value you assigned to the runtimeModeLimit setting (see "Adapter-Specific Settings" on page 42), the adapter user can select some combination of the following values in the Coordination Mode field as follows:

This runtimeModeLimit value...	Displays these values in the Coordination Mode field...
disable	disable
standby	disable and standby
distribute	disable, standby, and distribute

- The Max Process Time field enables other notifications to determine whether a currently executing notification should be considered to be "dead." If a notification executes a scheduled run and it fails to complete before the Max Process Time, then another notification instance will consider it dead; this other notification will assume control and execute a scheduled run. The default value is equal to the value in the watt.art.clusteredPollingNotification.keepAliveExpireTimeout setting in the server.cnf file.

If the Max Process Time setting is not high enough, you may encounter a situation in which a notification is executing normally but another notification assumes it is "dead." When the original notification completes, it will recognize that it was prematurely considered "dead." In this case, the system logs an Illegal Overlap exception with message id [ART.116.3715]. If this exception occurs, increase your Max Process Time setting.

When setting the value of Max Process Time, you should allow for "clock drift." For details, see "Clock Synchronization" on page 45.

If you want to update the schedule and settings of a notification in a cluster, all notification instances in the cluster must be suspended or disabled for the changes to be saved. If any notification instance in the cluster is enabled, the adapter will not save the updates.

If all instances of a notification in the cluster do not have the same settings, the notification that became active first will have precedence.

Clock Synchronization

To determine whether a notification has failed, notifications use the system clocks of the machines that host the clustered Integration Servers. Synchronizing the clocks of all machines in the cluster is critical for the proper execution of clustered polling notifications.

However, in time these clocks might become un-synchronized. Therefore you should anticipate the effect of "clock drift" when you establish values for the keepAliveExpireTimeout server-specific setting and Max Process Time notification-specific setting. Clock drift is the time difference between the clocks. As a guideline, add to the keepAliveExpireTimeout and the Max Process Time settings two times the maximum clock drift you anticipate.

Configuring Adapter Notification Schedules in a Clustered Environment

To enable Salesforce.com Adapter polling notifications to distribute or standby mode

1. In the cluster, shut down the Integration Server you are configuring.
2. Open the WmSalesforceAdapter\config\clusterProperties.cnf file.
3. Change all `<runtimeModeLimit>` value to `<distribute>` or `<standby>`. To see a sample clusterProperties.cnf file, see "Example Salesforce.com Adapter clusterProperties.cnf File" on page 43.
4. Save the file and restart the Integration Server.
5. Start the Integration Server Administrator.
6. Select Adapters > Salesforce.com Adapter.
7. From the navigation area, select Polling Notifications.

8 For each notification:

a Disable the notification.

b Click the Edit Schedule icon.

c Set the Coordination Mode to Distributed or Standby (as appropriate for the notification).

d Enable the notification.

Important! To maintain duplicate detection and ordering, your polling notification schedules must not run with the Overlap option selected. (To access the Overlap option, click the Edit Schedule icon.)

After you configure a polling notification, you may propagate all the affected components across your cluster. Changing the polling notification schedule from the Integration Server Administrator or changing the polling notification settings in Developer or Designer will require you to propagate the polling notification across the cluster. If you made changes to the settings in server.cnf or to the WmSalesforceAdapter\config\clusterProperties.cnf files, you must also propagate these across the cluster.

4 Adapter Connections

Overview

This chapter describes how to configure and manage Salesforce.com Adapter connections. For more information about how adapter connections work, see "Adapter Connections" on page 15.

Before Configuring or Managing Adapter Connections

To prepare to configure or manage an adapter connection

1 Install webMethods Integration Server and the Salesforce.com Adapter on the same machine. For details, see Chapter 2, "webMethods Salesforce.com Adapter Installation".

2 Make sure you have webMethods administrator privileges so that you can access the Salesforce.com Adapter's administrative screens. For information about setting user privileges, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7.

3 Start Integration Server and Integration Server Administrator, if they are not already running.

4 Using Integration Server Administrator, make sure the WmSalesforceAdapter package is enabled. See "Enabling Adapter Connections" on page 56 for instructions.

5 Using webMethods Developer or webMethods Designer, create a user-defined package to contain connections, if you have not already done so. See "Managing the Adapter Package" on page 34 for details.

6 Create your connections, as described in "Configuring Adapter Connections" on page 48.

Configuring Adapter Connections

When you configure Salesforce.com Adapter connections, you specify information that Integration Server uses to connect to Salesforce.com. You configure the connections using Integration Server Administrator.

To configure an adapter connection

1 In Integration Server Administrator select Adapters > Salesforce.com Adapter.

2 On the Connections screen, click Configure New Connection.

3 On the Connection Types screen, click Salesforce.com Connection to display the Configure Connection Type screen.

4 In the Salesforce.com Adapter Connection section, provide values for the following parameters:

Parameter	Description
Package	The package in which to create the connection. You must create the package using Developer or Designer before you can specify it using this parameter. For general information about creating packages, see the *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7. Note: Create the connection in a user-defined package rather than in the WmSalesforceAdapter package. See "Managing the Adapter Package" on page 34 for other important considerations when creating packages for the Salesforce.com Adapter.
Folder Name	The folder in which to create the connection.
Connection Name	The name you want to give the connection. Connection names cannot have spaces or use special characters reserved by Integration Server, Developer, or Designer. For more information about the use of special characters in package, folder, and element names, see the *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.

5 In the Connection Properties section, provide values for the following parameters:

Parameter	Description

Endpoint URL	Defines the login endpoint for initiating communications with Salesforce.com. The default URL for the WSDL version 13 is:
	
	`https://www.salesforce.com/services/Soap/u/13.0`
	
	The Salesforce.com Adapter enables users to specify an alternate URL when using the Salesforce.com Sandbox or a testing environment.
	
	To access the organization's Salesforce.com sandbox, specify the following URL:
	
	`https://test.salesforce.com/services/Soap/u/13.0`

User Name	Identifies the user name for the Salesforce.com user account that you want the adapter to access Salesforce.com. Typically, the user name is an e-mail address.

Parameter	Description
Password Retype Password	Identifies the password associated with User Name and the user's security token. For example, if the password is "mypassword" and the security token is "XXXXXXXXXX", specify the following for Password and Retype Password: `mypasswordXXXXXXXXXX` Salesforce.com requires a security token for access via the Force.com Web service API. To obtain the security token from the Salesforce.com user interface, select Setup; then in the My Personal Information section of the screen, click Reset your security token. Salesforce.com resets the security token and sends an e-mail message with the new security token to the registered user. Note: The security token you specify in the connection is valid until you reset it, change your password, or have your password reset. The security token is not required, if the Endpoint URL is a Salesforce.com sandbox.
Connector Timeout (sec)	Defines the number of seconds that Salesforce.com Adapter waits for a response from Salesforce.com before reporting a connection timeout. This is a SOAP connection timeout. The default is 60 seconds. The adapter expects a positive integer value. A value of 0 means no timeout.

Parameter	Description
Session Timeout (min)	Defines the number of minutes to maintain a Salesforce.com user session before performing a re-login to keep the session alive. The default is 30 minutes. Specify the same value here as in the Security Controls setup area in the Salesforce.com user interface, to allow the adapter to automatically re-login to prevent a session from expiring. After a session is established, Salesforce.com automatically expires the session after the number of minutes specified in the Security Controls setup area in the Salesforce.com user interface.
Use Compression	Indicates whether you want the adapter to enable compression in the SOAP request and response messages sent to Salesforce.com. ■ Select **false** to disable compression. This is the default. ■ Select **true** to enable compression. As defined by the HTTP 1.1 specification to indicate that compression is supported, the adapter includes "Accept-Encoding: gzip, deflate" in the HTTP header of the request message. For compressed response messages, the HTTP header includes "Content-Encoding: deflate" or "Content-Encoding: gzip", as appropriate.

6 In the **Connection Management Properties** section, provide values for the following parameters:

Parameter	Description
Enable Connection Pooling	Enables the adapter to use connection pooling. The default is `true`. See "Connection Pools" on page 15 for more information about connection pooling in the adapter. If you plan to enable connection pooling in a clustered environment, consider the connection pool size. For details, see "Considerations When Configuring Connections with Connection Pooling Enabled" on page 41.
Minimum Pool Size	The minimum number of connection objects that remain in the connection pool at all times. When the adapter creates the pool, it creates this number of connections. The default is `1`.

Parameter	Description
Maximum Pool Size	The maximum number of connection objects that can exist in the connection pool. When the connection pool has reached its maximum number of connections, the adapter will reuse any inactive connections in the pool or, if all connections are active, it will wait for a connection to become available. The default is `10`.
Pool Increment Size	If connection pooling is enabled, this parameter specifies the number of connections by which the pool will be incremented if connections are needed, up to the maximum pool size. The default is `1`.
Block Timeout	If connection pooling is enabled, this parameter specifies the number of milliseconds that Integration Server will wait to obtain a connection before it times out and returns an error. The default is `1000`.
Expire Timeout	If connection pooling is enabled, this parameter specifies the number of milliseconds that an inactive connection can remain in the pool before it is closed and removed from the pool. For example, to specify 10 seconds, specify `10000`. Enter `0` to specify no timeout. The default is `1000`. Note: The adapter will never violate the Minimum Pool Size parameter. These connections remain in the pool regardless of how long they are inactive.
Startup Retry Count	If connection pooling is enabled, this parameter specifies the number of times that the system should attempt to initialize the connection pool at startup if the initial attempt fails, before issuing an AdapterConnectionException. The default is `0`.
Startup Backoff Timeout	If connection pooling is enabled, this parameter specifies the number of seconds to wait between each attempt to initialize the connection pool. The default is `10`.

7 Click Save Connection.

The connection you created appears on the adapter's Connections screen and in the Developer's Service Browser.

8 Be sure to enable the connection before you create adapter services that use it. See "Enabling Adapter Connections" on page 56 for instructions.

Dynamically Changing a Service's Connection at Run Time

You can run an adapter service using a connection other than the default connection that was associated with the service when it was created. To override the default, you must code your flow service to pass a value through the pipeline into the adapter service's *$connectionName* field.

For example, you have a flow service that creates a new account for the Account Salesforce object in the Salesforce.com production area. However, for testing purposes you want the flow service to have the capability to create the account in the Salesforce.com Sandbox. At run time, you want the flow service to determine programmatically whether to create the account in the production area or the Sandbox. Set up your flow service to have a *Target* variable so that it can branch its logic based on the value in *Target*:

- If *Target* contains the value `production`, the flow service does not specify a value for the *$connectionName* pipeline variable, which causes the adapter service to use its default connection to the production area.
- If *Target* contains the value `test`, the flow services sets the *$connectionName* variable to the connection for the Sandbox so that the adapter service creates the account in the Sandbox.

Alternatively, you can use the configured connection for an adapter service, but at run time override the Salesforce.com user credentials defined in the connection. To override the user credentials, you must code your flow service to pass values through the pipeline into the adapter service's *username* and *password* fields.

For more information, see "Changing the Connection Associated with an Adapter Service at Run Time" on page 17.

Viewing Adapter Connection Parameters

You can view a connection's parameters from Integration Server Administrator, Developer, or Designer.

To view the parameters for an adapter connection using the Integration Server Administrator

1 In the Adapters menu in the Integration Server Administrator's navigation area, click Salesforce.com Adapter.

When using the adapter with Integration Server 8.0 and later, you can sort and filter the list of connections that appears on the Connections screen.

- To sort information on the Connections screen, click the Up and Down arrows at the top of the column you want to sort.

- To filter the list of connections:

  1 On the Connections screen, click Filter Connections.

  2 Type the criterion by which you want to filter into the Filter criteria box. Filtering is based on the node name, not 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.

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

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

  The Connections screen appears, listing all the current connections. You can control the number of connections that are displayed on this screen. For more information, see "Controlling Pagination" on page 25.

2 On the Connections screen, click  for the connection you want to see.

  The View Connection screen displays the parameters for the connection. For descriptions of the connection parameters, see the table of parameters in "Configuring Adapter Connections" on page 48.

3 Click Return to Salesforce.com Adapter Connections to return to the Connections screen.

To view the parameters for an adapter connection using Developer or Designer

1 From the Developer or Designer navigation area, open the package and folder in which the connection is located.

2 Double-click the connection you want to view.

  The parameters for the connection appear in the Connection Information tab. For descriptions of the connection parameters, see the table of parameters in "Configuring Adapter Connections" on page 48.

Editing Adapter Connections

Use Integration Server Administrator to redefine parameters for a connection.

To edit an adapter connection

1 Disable the connection you want to edit. See "Disabling Adapter Connections" on page 56 for instructions.

2 In Integration Server Administrator, select Adapters > Salesforce.com Adapter.

3 On the Connections screen, click  for the connection you want to edit.

In the Edit Connection screen, update the parameters by typing or selecting the values you want to specify. For descriptions of the connection parameters, see "Configuring Adapter Connections" on page 48.

4 Click Save Changes to save the connection and return to the Connections screen.

5 Enable the connection you edited. See "Enabling Adapter Connections" on page 56 for instructions.

Copying Adapter Connections

You can copy an existing Salesforce.com Adapter connection to create a new connection with the same or similar connection properties without retyping all properties for the new connection.

To copy an adapter connection

1 In Integration Server Administrator, select Adapters > Salesforce.com Adapter.

2 On the Connections screen, click  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 and edit any connection parameters as needed by typing or selecting the values you want to specify. For descriptions of the connection parameters, see "Configuring Adapter Connections" on page 48.

3 Click Save Connection Copy to save the connection and return to the Connections screen.

Deleting Adapter Connections

You can delete Salesforce.com Adapter connections that you no longer want to use.

When you delete a Salesforce.com Adapter connection, the adapter services that are defined to use the connection will no longer work. To assign a different connection to an adapter service and re-use the service, use the webMethods setAdapterServiceNodeConnection built-in service. For more information, see "Changing the Connection Associated with an Adapter Service at Design Time" on page 17.

To delete an adapter connection

1 Disable the connection you want to delete. See "Disabling Adapter Connections" on page 56 for instructions.

2 In Integration Server Administrator, select Adapters > Salesforce.com Adapter.

3 On the Connections screen, click  for the connection you want to delete.

Integration Server deletes the adapter connection.

Enabling Adapter Connections

Adapter connections must be enabled before you can create adapter services or adapter notifications for those connections.

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 an adapter connection

1 In Integration Server Administrator, select Adapters > Salesforce.com Adapter.

2 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  and Yes in the Enabled column.

Disabling Adapter Connections

Adapter connections must be disabled before you can edit or delete the connections.

To disable an adapter connection

1 In Integration Server Administrator, select Adapters > Salesforce.com Adapter.

2 On the Connections screen, click Yes in the Enabled column for the connection you want to disable.

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

5 Adapter Services

Overview

The following sections describe how to configure adapter services, which you use for webMethods to Salesforce.com communications. All adapter services work on a single Salesforce.com object at a time.

Before you configure Salesforce.com Adapter services, you must configure the connections you plan to use with them. For details, see "Configuring Adapter Connections" on page 48.

Before Configuring or Managing Adapter Services

To prepare to configure or manage an adapter service

1 Start your Integration Server and the Integration Server Administrator, if they are not already running.

2 Make sure you have Integration Server administrator privileges so that you can access the Salesforce.com Adapter administrative screens. For information about setting user privileges, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7.

3 Using the Integration Server Administrator, make sure the WmSalesforceAdapter package is enabled. For instructions, see "Enabling and Disabling Packages" on page 36.

4 Using the Integration Server Administrator, configure an adapter connection to use with the adapter service. For instructions, see "Configuring Adapter Connections" on page 48.

5 Start Developer or Designer if it is not already running.

> Note: If you are using Developer 6.1 or later, use the Edit perspective for all procedures unless stated otherwise. If you are using Designer, use the Service Development perspective. For more information, see the *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.

6 Using Developer or Designer, create a package to contain adapter services, if you have not already done so. When you configure adapter services, define them in user-defined packages instead of in the WmSalesforceAdapter package. For more information about managing packages for the adapter, see Chapter 3, "Adapter Package Management".

Configuring Adapter Services

You can configure adapter services that perform the following operations:

Operation	For more information, see...
Insert new records into a Salesforce object.	"Configuring Create Operation Adapter Services" on page 60
Delete records from an organization's data.	"Configuring Delete Operation Adapter Services" on page 62
Perform a query against a Salesforce object to return field values from matching records.	"Configuring Query Operation Adapter Services" on page 64
Retrieve field values of specified records of a Salesforce object. Use this operation when you know the IDs of the records from which you want field values.	"Configuring Retrieve Operation Adapter Services" on page 67
Update records of a Salesforce object.	"Configuring Update Operation Adapter Services" on page 70
Perform an upsert operation to either insert new records into or update records of a Salesforce object.	"Configuring Upsert Operation Adapter Services" on page 72
Execute the Salesforce.com getServerTimestamp utility.	"Configuring Utility Operation Adapter Services" on page 75
Execute the Salesforce.com getUserInfo utility.	"Configuring Utility Operation Adapter Services" on page 75
Execute the Salesforce.com resetPassword utility.	"Configuring Utility Operation Adapter Services" on page 75
Execute the Salesforce.com sendEmail utility.	"Configuring Utility Operation Adapter Services" on page 75
Execute the Salesforce.com sendMassEmail utility.	"Configuring Utility Operation Adapter Services" on page 75
Execute the Salesforce.com setPassword utility.	"Configuring Utility Operation Adapter Services" on page 75

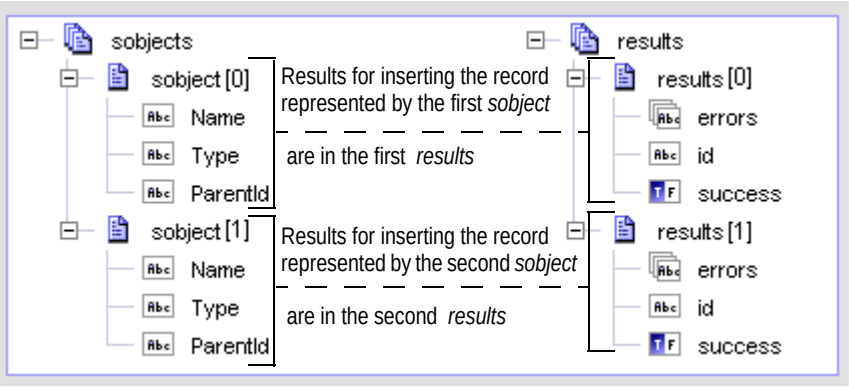
Configuring Create Operation Adapter Services

Use the Create Operation template to configure an adapter service that inserts new records into a Salesforce object. For example, you might configure an adapter service that inserts new contacts into the Contact Salesforce object. The Create operation wraps the create call of the Salesforce.com Web service API.

The Salesforce.com create call returns a SaveResult object. The adapter service returns the information from the SaveResult object as output.

Output Variable	Description
id	String The ID of a newly inserted record.
success	Boolean Whether the create call successfully inserted a record.
errors	String List If *success* is `false`, *errors* provides the error codes and descriptions of the errors that the create call encountered.

The input signature of the adapter service uses an *sobjects* Document List, which holds the field values for the records to insert. At run time, pass the adapter service one Document within the *sobjects* Document List for each record you want to insert. The output signature of the adapter service contains a *results* Document List. Each Document in the input *sobjects* Document List maps to an output Document in the *results* Document List.



To configure an adapter service that uses the Create Operation template

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 Start Developer or Designer.

3 Perform one of the following:

- If you are using Developer, select File > New > Adapter Service and click Next.
- If you are using Designer, perform the following:

1. Right-click the package in which the service should be contained and select New > Adapter Service.
2. Select the parent namespace and type a name for the adapter service.
3. Click Next.

4. Select Salesforce.com Adapter as the adapter type and click Next.
5. Select the appropriate Adapter Connection Name and click Next.

Important! Ensure that the Salesforce.com user account identified by the connection has all the required permissions to insert a record in the Salesforce object that you will select in step 7. See "Configuring Adapter Connections" on page 48 for more information.

---

6. From the list of available templates, select the Create Operation template and do one of the following:
   - If you are using Developer, click Next, select a package and folder to contain the service, type a unique name for the service, and click Finish.
   - If you are using Designer, click Finish.

   The adapter service editor for the adapter service appears. You can select the Adapter Settings tab at any time to confirm adapter service properties such as the Adapter Name, Adapter Connection Name, and Adapter Service Template, as necessary.
7. Identify the Salesforce object in which the adapter service is to insert records.
   a. Select the Schema tab.
   b. From the Salesforce object list, select the Salesforce object into which you want to insert one or more records.
8. Identify the fields to include in the new record.
   a. Select the Create Operation tab.
   b. Click the  icon to populate the table with the creatable fields of the selected Salesforce object.
   c. For each field that you want to include in the new record, select the check box in the Use Field column. These are the fields to supply for the records being inserted.

      For example, if you are defining an adapter service that inserts records into the Account Salesforce object, you might define the adapter service to provide values for the Account Name, Account Type, and Parent Account ID fields. The fields become part of the input signature for the adapter service. When invoking the adapter service, pass the values for these fields as input. The adapter service uses those input values when creating the new records.

Important! Be sure to select all required fields for a record and all fields required to maintain the referential integrity of the Salesforce object.

Note: The **Create Operation** template does not list fields that are automatically created when a new record is added, for example, CreatedDate, CreatedById, LastModifiedDate, LastModifiedById, and SystemModstamp. Although you cannot select these fields, they will be included in new records. Also, the **Create Operation** template shows only those fields that are marked "creatable" for the create operation.

At run time, the adapter service determines whether the values specified for the fields are valid, for example ensuring that a date value is specified for a field that has a java.util.Date data type. If an invalid value is detected, the adapter issues an error message and does not attempt to insert the new records.

9 Select **File > Save** to save the adapter service.

After you save the adapter service, Developer or Designer updates the input signature on the **Input/Output** tab to include the fields you selected.

Configuring Delete Operation Adapter Services

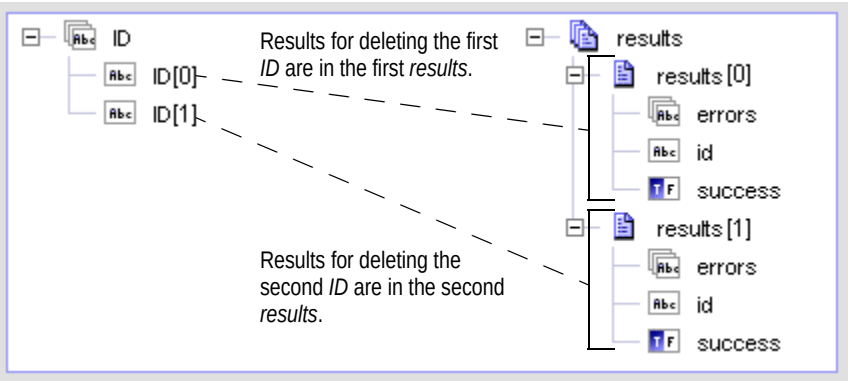
Use the **Delete Operation** template to configure an adapter service that deletes records from your organization's data. For example, you might configure an adapter service that deletes contacts from the Contact Salesforce object. The Delete operation wraps the delete call of the Salesforce.com Web service API.

The input to the adapter service includes an *ID* String List. At run time, pass the adapter service the ID of each record to delete. Each ID is a String value within the *ID* String List.

As a response, the Salesforce.com delete call returns a DeleteResult object. The adapter service returns the information from the DeleteResult object as output.

Output Variable	Description
id	**String** The ID of a deleted record.
success	**Boolean** Whether the delete call successfully deleted a record.
errors	**String List** If *success* is `false`, *errors* provides the error codes and descriptions of the errors that the delete call encountered.

The IDs of the records to delete in the *ID* String List input parameter map to the output in the *results* Document List.



To configure an adapter service that uses the Delete Operation template

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 Start Developer or Designer.

3 Perform one of the following:

- If you are using Developer, select File > New > Adapter Service and click Next.
- If you are using Designer, perform the following:

  1 Right-click the package in which the service should be contained and select New > Adapter Service.

  2 Select the parent namespace and type a name for the adapter service.

  3 Click Next.

4 Select Salesforce.com Adapter as the adapter type and click Next.

5 Select the appropriate Adapter Connection Name and click Next.

Important! Ensure the Salesforce.com user account identified by the connection has all the required permissions to delete records from the Salesforce object.

6 From the list of available templates, select the Delete Operation template and do one of the following:

- If you are using Developer, click Next, select a package and folder to contain the service, type a unique name for the service, and click Finish.
- If you are using Designer, click Finish.

The adapter service editor for the adapter service appears. You can select the Adapter Settings tab at any time to confirm adapter service properties such as the Adapter Name, Adapter Connection Name, and Adapter Service Template, as necessary.

7 Select File > Save to save the adapter service.

Configuring Query Operation Adapter Services

Use the Query Operation template to configure an adapter service that executes a Salesforce Object Query Language (SOQL) query against a Salesforce object, to find records and return selected field values from those records. For example, you might configure an adapter service that queries the Contact Salesforce object to find contacts who have a specific Title and return values for the FirstName, LastName, and Phone fields.

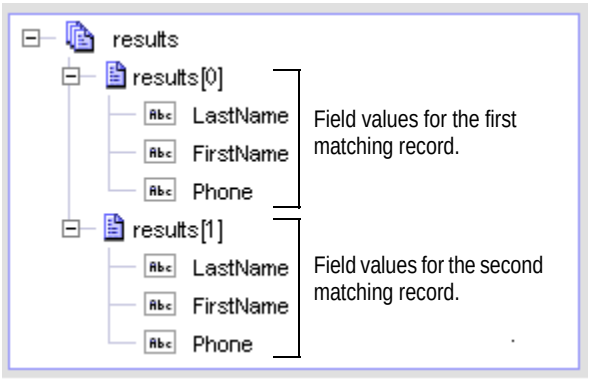
The Query operation wraps the query and queryMore calls of the Salesforce.com Web service API. The query call executes the initial query to retrieve the initial matching records. If the response from the query call indicates that there are additional records to retrieve, the adapter service executes the queryMore call as many times as necessary to retrieve all records.

At run time when you invoke the service, pass the input parameters to the adapter service if you have specified variables in the WHERE clause. The adapter service generates the query and executes the query.

As a response, the Salesforce.com query and queryMore calls return a QueryResult object, which includes the selected field values of the matching records and a flag indicating whether there are additional matching records to retrieve. The adapter service combines the results from all query and queryMore calls.

The following table describes the output from a Query Operation adapter service:

Variable	Description
results	Document List Each Document in the Document List contains the selected fields for a record that matched the specified query.



To configure an adapter service that uses the Query Operation template

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 Start Developer or Designer.

3 Perform one of the following:

- If you are using Developer, select File > New > Adapter Service and click Next.
- If you are using Designer, perform the following:
  1. Right-click the package in which the service should be contained and select New > Adapter Service.
  2. Select the parent namespace and type a name for the adapter service.
  3. Click Next.

4. Select Salesforce.com Adapter as the adapter type and click Next.
5. Select the appropriate Adapter Connection Name and click Next.

Important! Ensure the Salesforce.com user account identified by the connection has all the required permissions to query records from the Salesforce object that you will select in step 7.

---

6. From the list of available templates, select the Query Operation template and do one of the following:
   - If you are using Developer, click Next, select a package and folder to contain the service, type a unique name for the service, and click Finish.
   - If you are using Designer, click Finish.

   The adapter service editor for the adapter service appears. You can select the Adapter Settings tab at any time to confirm adapter service properties such as the Adapter Name, Adapter Connection Name, and Adapter Service Template, as necessary.
7. Identify the Salesforce object that you want to query.
   a. Select the Schema tab.
   b. From the Salesforce object list, select the Salesforce object you want to query.
8. Identify the fields of the matching records that you want returned. If you do not specify fields, the adapter throws an error stating MALFORMED_QUERY.
   a. Select the Query Operation tab.
   b. Click the  icon to populate the table with the fields of the selected Salesforce object.
   c. For each field you want returned, select the check box in the Use Field column.

      For example, if you are querying the Account Salesforce object, you may select to have the adapter service return the “Account ID” and “Account Name” fields of the matching records.

9 Optionally, define how you want the adapter to sort the retrieved records.

a For each field you want to sort by, in the Sort Order column select a sort order option. Only specify a sort option for fields you select using the Use Field check box. The adapter ignores sort options that you specify for non-selected fields.

Select this option...	To sort the results in...
ASC NULLS FIRST	Ascending order by the field value, with the null values at the beginning of the results.
ASC NULLS LAST	Ascending order by the field value, with the null values at the end of the results.
DESC NULLS FIRST	Descending order by the field value, with the null values at the beginning of the results.
DESC NULLS LAST	Descending order by the field value, with the null values at the end of the results.

b If you are sorting by multiple fields, move the rows into the order in which you want the adapter to perform the sorts. The adapter performs each sort in the order the rows are listed on the Query Operation tab.

To move a row, select it and then click the  icon to move the row up, or click the  icon to move the row down.

For more information about sorting, including factors that affect sorting and data type limitations, see the *Force.com Web Services API Developer's Guide*.

10 Optionally, you can use the WHERE field to filter the search results for an object based on specific field values. You specify the filter using Salesforce Object Query Language (SOQL). If you leave the WHERE field blank, the adapter returns all records in the Salesforce object specified on the Schema tab.

You can either specify the WHERE condition or you can use variables. When you use variables in a WHERE clause, use ${*INPUT_FIELD_NAME*} to map a part of the clause to the input field. At design time, the service template generates an input field with INPUT_FIELD_NAME, which allows you to specify placeholders within the WHERE clause (conditional Expression) free text area to define input parameters. At run time, the service parses the WHERE clause and replaces the ${*INPUT_FIELD_NAME*} with the actual contents of the input field.

For example, consider the following WHERE clauses:

- `Name = '${name}' and Sales > 200000`

  In this example, the service template will generate an input field for *name*. At run time, `${name}` is replaced with the value of *name* variable.

  If you specify `Jack` as the value for the name input field, then the resultant WHERE clause is:

  ```
  Name = 'Jack' and Sales > 200000
  ```

- `Name = '${name}' and Sales > ${amount}`

  In this example, the service template will generate input fields for ***name*** and ***amount*** .

- `${where}`

  In this example, the service template will generate an input field for ***where***. In the ***where*** input field, you can specify any condition.

- `Type = 'Customer' or ${condition}`

  In this example, the service template will generate an input field for ***condition***. In the ***condition*** input field, you can specify the dynamic condition.

For more information about SOQL and the WHERE conditionExpression, see the *Force.com Web Services API Developer's Guide*.

11 In the Maximum Rows field, type a positive, whole number to specify the maximum number of records that you want returned. If you want all records to be returned, type `0`.

12 Select File > Save to save the adapter service.

Configuring Retrieve Operation Adapter Services

Use the Retrieve Operation template to configure an adapter service that retrieves selected fields from records of a Salesforce object. The Retrieve operation wraps the retrieve call of the Salesforce.com Web service API.

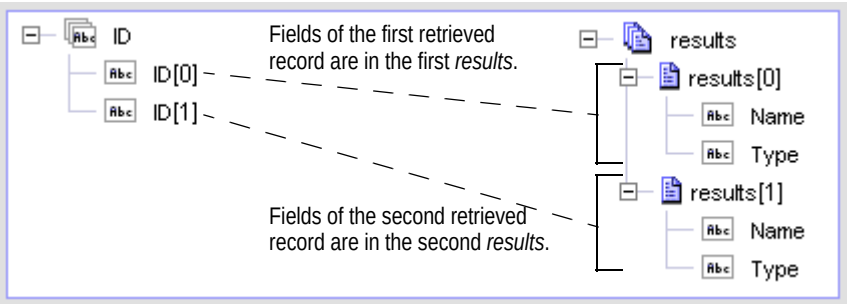
The input to the adapter service includes an *ID* String List. At run time, pass the adapter service the ID of each record to retrieve. Each ID is a String value within the *ID* String List.

As a response, the Salesforce.com retrieve call returns the objects that contain the field values of the retrieved records.

The following table describes the output from a Retrieve Operation adapter service:

Variable	Description
results	Document List Each Document in the Document List contains the selected fields of the retrieved records. There is a one-to-one mapping between the IDs of the records specified in the ID input parameter to the output in the results Document List.
invalidIds	String List Contains the list of invalid IDs. If the user does not have sufficient privilege or if one or more IDs specified in the input parameter are invalid, the response from the Salesforce.com retrieve() call is null. The Salesforce.com Adapter cannot retrieve the records for invalid IDs. These invalid IDs are made a part of the *invalidIds* output parameter. Note: When there are invalid IDs in the *invalidIds* output parameter, the number of records in the *results* output parameter will not match the number of IDs specified in the input signature.

The IDs of the records to retrieve in the *ID String List* input parameter map to the output in the *results* Document List.



Note: Salesforce.com imposes restrictions on the number of IDs that can be passed for a single call. Ensure that the number of IDs passed to a single Retrieve operation does not exceed the maximum limit. For more information, see the *Force.com Web Services API Developer's Guide*.

To configure an adapter service that uses the Retrieve Operation template

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 Start Developer or Designer.

3 Perform one of the following:

- If you are using Developer, select File > New > Adapter Service and click Next.

- If you are using Designer, perform the following:
  1. Right-click the package in which the service should be contained and select New > Adapter Service.
  2. Select the parent namespace and type a name for the adapter service.
  3. Click Next.

4. Select Salesforce.com Adapter as the adapter type and click Next.
5. Select the appropriate Adapter Connection Name and click Next.

Important! Ensure the Salesforce.com user account identified by the connection has all the required permissions to retrieve records from the Salesforce object that you will select in step 7.

---

6. From the list of available templates, select the Retrieve Operation template and do one of the following:
   - If you are using Developer, click Next, select a package and folder to contain the service, type a unique name for the service, and click Finish.
   - If you are using Designer, click Finish.

   The adapter service editor for the adapter service appears. You can select the Adapter Settings tab at any time to confirm adapter service properties such as the Adapter Name, Adapter Connection Name, and Adapter Service Template, as necessary.
7. Identify the Salesforce object from which the adapter service is to retrieve records.
   a. Select the Schema tab.
   b. From the Salesforce object list, select the Salesforce object from which you want to retrieve records.
8. Identify the fields of the retrieved records that you want returned. If you do not specify fields, the adapter reports an INVALID_FIELD error, which Salesforce.com returns.
   a. Select the Retrieve Operation tab.
   b. Click the  icon to populate the table with the fields of the selected Salesforce object.
   c. For each field want returned, select the check box in the Use Field column.
9. Select File > Save to save the adapter service.

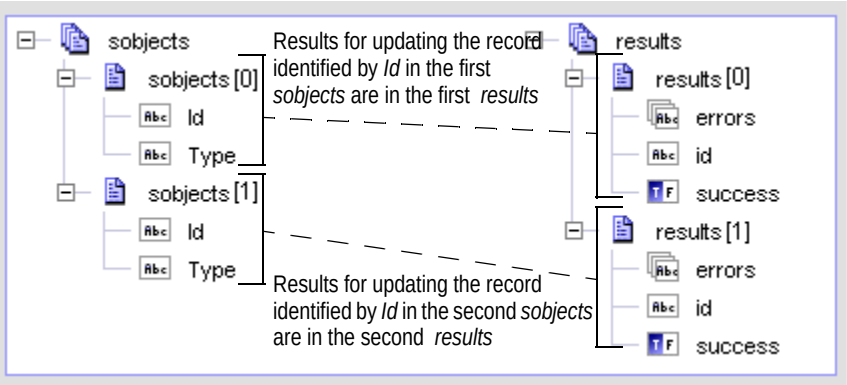
Configuring Update Operation Adapter Services

Use the Update Operation template to configure an adapter service that updates records of a Salesforce object. The update operation wraps the update call of the Salesforce.com Web service API.

As a response, the Salesforce.com update call returns a SaveResult object. The adapter service returns the information from the SaveResult object as output.

Output Variable	Description
id	String The ID of an updated record.
success	Boolean Whether the update call successfully updated a record.
errors	String List If *success* is `false`, *errors* provides the error codes and descriptions of the errors that the update call encountered.

The input signature of the adapter service uses an *sobjects* Document List, which holds the ID of a record to update and the values for the fields to update. At run time, pass the adapter service one Document within the *sobjects* Document List for each record you want to update. The output signature of the adapter service contains a *results* Document List. Each Document in the input *sobjects* Document List maps to an output Document in the *results* Document List.



To configure an adapter service that uses the Update Operation template

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 Start Developer or Designer.

3 Perform one of the following:

- If you are using Developer, select File > New > Adapter Service and click Next.
- If you are using Designer, perform the following:

1. Right-click the package in which the service should be contained and select New > Adapter Service.
2. Select the parent namespace and type a name for the adapter service.
3. Click Next.

4. Select Salesforce.com Adapter as the adapter type and click Next.
5. Select the appropriate Adapter Connection Name and click Next.

Important! Ensure the Salesforce.com user account identified by the connection has all the required permissions to update records in the Salesforce object that you will select in step 7.

---

6. From the list of available templates, select the Update Operation template and do one of the following:
   - If you are using Developer, click Next, select a package and folder to contain the service, type a unique name for the service, and click Finish.
   - If you are using Designer, click Finish.

   The adapter service editor for the adapter service appears. You can select the Adapter Settings tab at any time to confirm adapter service properties such as the Adapter Name, Adapter Connection Name, and Adapter Service Template, as necessary.
7. Identify the Salesforce object that contains the records you want to update.
   a. Select the Schema tab.
   b. From the Salesforce object list, select the Salesforce object that contains the records that you want to update.
8. Select the Id field so that it is included in the input parameters, providing a variable for passing the IDs of the records that the adapter service is to update.
   a. Select the Update Operation tab.
   b. Click the  icon to add one field row to the table.
   c. If the newly added field is not the Id field, select Id from the list in the Input Parameter column.
   d. Select the check box in the Use Field column for the Id field.
9. Identify the fields of the records that you want to update.
   a. Select the Update Operation tab.
   b. Click the  icon to populate the table with the rest of the fields of the selected Salesforce object.
   c. For each field want to update, select the check box in the Use Field column.

For example, if you are defining an adapter service that updates records in the Contact Salesforce object, you might set up the adapter service to update the "Email", "Phone", and "MobilePhone" fields for a contact. The fields you specify become part of the input signature for the adapter service. When invoking the adapter service, pass the values for the selected fields as input. To reset a non-required field to null, pass a null value for the field.

At run time, the adapter service determines whether the values specified for the fields are valid, for example ensuring that a date value is specified for a field that has a java.util.Date data type. If an invalid value is detected, the adapter issues an error message and does not attempt to update the record.

Important! When using an update operation adapter service, be sure to specify field values that maintain the referential integrity of the Salesforce object.

Note: The Update Operation template lists only fields that you can update for a Salesforce object. If Salesforce.com does not allow you to update a field, for example the MasterRecordId field for a Contact Salesforce object, the adapter service template does not list the field for selection.

---

10 Select File > Save to save the adapter service.

Configuring Upsert Operation Adapter Services

Use the Upsert Operation template to configure an adapter service that inserts records or updates records of a Salesforce object. When attempting to add new records, you can use the upsert operation instead of the create operation to avoid the potential of inserting duplicate records. The upsert operation wraps the upsert call of the Salesforce.com Web service API.

For the upsert operation, you specify the value of a field that the operation uses to determine whether a record already exists. The field must be:

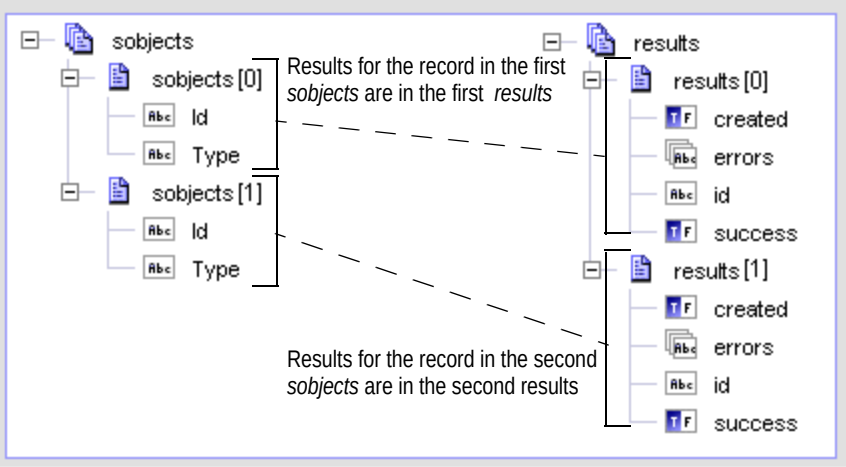
- For standard objects, a field with the idLookup property. This is usually the object's ID, Name field, or both.
- For custom objects, a field with the external ID attribute.

If the record does not already exist, the upsert operation inserts it. If the record does exist, the upsert operation updates it. You cannot create or update the values of the fields, which are auto generated by Salesforce.com.

As a response, the Salesforce.com upsert call returns a UpsertResult object. The adapter service returns the information from the UpsertResult object as output.

Output Variable	Description
created	Boolean Whether the upsert operation inserted or updated the specified record. The value of *created* is `true` if the upsert operation inserted the record or `false` if the upsert operation updated the record.
id	String The ID of the created or updated record when *success* is `true`. If *success* is `false`, *id* is null.
success	Boolean Whether the upsert call successfully inserted/updated the record.
errors	String List If *success* is `false`, *errors* provides the error codes and descriptions of the errors that the upsert call encountered.

The input signature of the adapter service uses an *sobjects* Document List, which holds the field values the upsert operation uses. At run time, pass the adapter service one Document within the *sobjects* Document List for each record on which you want to act. The output signature of the adapter service contains a *results* Document List. Each Document in the input *sobjects* Document List maps to an output Document in the *results* Document List.



To configure an adapter service that uses the Upsert Operation template

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 Start Developer or Designer.

3 Perform one of the following:

- If you are using Developer, select File > New > Adapter Service and click Next.
- If you are using Designer, perform the following:
  1. Right-click the package in which the service should be contained and select New > Adapter Service.
  2. Select the parent namespace and type a name for the adapter service.
  3. Click Next.

4. Select Salesforce.com Adapter as the adapter type and click Next.
5. Select the appropriate Adapter Connection Name and click Next.

Important! Ensure the Salesforce.com user account identified by the connection has all the required permissions to insert/update records in the Salesforce object that you will select in step 7.

---

6. From the list of available templates, select the Upsert Operation template and do one of the following:
   - If you are using Developer, click Next, select a package and folder to contain the service, type a unique name for the service, and click Finish.
   - If you are using Designer, click Finish.

   The adapter service editor for the adapter service appears. You can select the Adapter Settings tab at any time to confirm adapter service properties such as the Adapter Name, Adapter Connection Name, and Adapter Service Template, as necessary.
7. Identify the Salesforce object on which you want the adapter service to act and the field to use to determine whether a record already exists.
   a. Select the Schema tab.
   b. From the Salesforce object list, select the Salesforce object for which you want to insert/update records.

      Developer updates the ID Field Name field with selections based on the Salesforce object that you select.
   c. From the ID Field Name list, select the field that you want to use to determine whether a record already exists in the Salesforce object.

      At run time, pass a value for this field as input into the adapter service. If the value matches an existing record, the adapter service updates the record. If the value does not match an existing record, the adapter service inserts a new record.
8. Include the ID Field Name that you selected in step 7 to the input parameters to provide a variable for passing the value that the upsert operation uses to determine whether a record exists.
   a. Select the Upsert Operation tab.
   b. Click the  icon to add one field row to the table.

c If the newly added field is not the field you specified for ID Field Name, select that field from the list in the Input Parameter column.

d Select the check box in the Use Field column for the field.

Note: If you do not include the ID Field Name field on the Upsert Operation tab, the upsert operation will insert new records, generating IDs for each record. If you do include the ID Field Name field and a value supplied does not match an existing record, the upsert operation inserts the new record using the value you supply for the field.

9 Identify the fields of the records for which you want to supply information.

a Select the Upsert Operation tab.

b Click the  icon to populate the table with the fields of the selected Salesforce object. These are the fields that you can update or create with the adapter service.

c For each field that you want to supply, select the check box in the Use Field column.

Important! When using an upsert operation adapter service, be sure to specify field values that maintain the referential integrity of the Salesforce object.

Note: The Upsert Operation template lists only fields that can be created and updated.

If the adapter service inserts a record, these are the fields the adapter service supplies for the new record. If the adapter service updates a record, these are the fields the adapter service updates. The fields you specify become part of the input signature for the adapter service.

At run time, the adapter service determines whether the values specified for the fields are valid, for example, ensuring that a date value is specified for a field that has a java.util.Date data type. If an invalid value is detected, the adapter issues an error message and does not attempt to insert/update the record.

10 Select File > Save to save the adapter service.

Configuring Utility Operation Adapter Services

Use the Utility Operation template to configure an adapter service that invokes one of the following Salesforce.com utilities:

- getServerTimestamp to retrieve the current Salesforce.com system timestamp.
- getUserInfo to obtain personal information about the currently logged-in user, that is, the user account used for the adapter connection. Your adapter service might obtain this information to use for display purposes.
- resetPassword to change a user's password to a temporary, system-generated value.

- sendEmail to send a single e-mail message.
- sendMassEmail to send multiple e-mail messages.
- setPassword to change the password of a user's password to a specified value.

You select the utility that you want the adapter service to execute. The input and output signatures of the adapter service are based on the inputs and outputs of the selected utility. For more information about utilities and parameters that a utility expects as input or returns as output, see the *Force.com Web Services API Developer's Guide*.

To configure an adapter service that uses the Utility Operation template

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 Start Developer or Designer.

3 Perform one of the following:

- If you are using Developer, select File > New > Adapter Service and click Next.
- If you are using Designer, perform the following:
  1 Right-click the package in which the service should be contained and select New > Adapter Service.
  2 Select the parent namespace and type a name for the adapter service.
  3 Click Next.

4 Select Salesforce.com Adapter as the adapter type and click Next.

5 Select the appropriate Adapter Connection Name and click Next.

> Important! Ensure the Salesforce.com user account identified by the connection has all the required permissions to execute the utility that you will select in step 7.

6 From the list of available templates, select the Utility Operation template and do one of the following:

- If you are using Developer, click Next, select a package and folder to contain the service, type a unique name for the service, and click Finish.
- If you are using Designer, click Finish.

The adapter service editor for the adapter service appears. You can select the Adapter Settings tab at any time to confirm adapter service properties such as the Adapter Name, Adapter Connection Name, and Adapter Service Template, as necessary.

7 Identify the utility that you want the adapter service to invoke.

a Select the Utility Operation tab.

b From the Utility Call list, select the utility you want to use.

8 View the Input Fields tab to see the inputs for the utility and the Output Fields tab to see the outputs that the utility returns. For a description of the inputs and output fields, see the *Force.com Web Services API Developer's Guide*.

9 Select File > Save to save the adapter service.

Testing Adapter Services

You can test adapter services using either Developer or Designer. For more information about testing and debugging services, see the *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.

To test an adapter service

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 If you are using Developer, set the view to the Test perspective.

3 In the Developer or Designer, expand the package and folder that contain the service you want to test.

4 Double-click the service you want to test.

Developer or Designer displays the configured service in the service template's Adapter Service Editor.

5 Perform one of the following

- If you are using Developer, select Test > Run.
- If you are using Designer, select Run > Run As > Run Service.

6 Optionally, provide the user credentials to connect to Salesforce.com. This user name and password overrides credentials specified in the adapter connection being used for the adapter service.

Parameter	Description/Action
username	The user name used to connect to the Salesforce.com.
password	The password used to connect to the Salesforce.com. Salesforce.com requires a security token for access via Force.com Web service API. You can obtain the security token from the Salesforce.com user interface. For example, if the password is "mypassword" and the security token is "XXXXXXXXXX", specify the following for Password and Retype Password: `mypasswordXXXXXXXXXX` Note: The security token is valid until you reset it, change your password, or have your password reset. The security token is not required, if the Endpoint URL is a Salesforce.com sandbox.

7 Specify values for the input parameters of the adapter service.

8 Optionally, specify an alternative connection to use by providing the connection name in the *$connectionName* input parameter.

Note: Credentials you provide in the *username* and *password* fields override any connection. Specifying the *$connectionName* input parameter changes the connection for this execution of the service. To reconfigure the service to use a different connection, use the setAdapterServiceNodeConnection, which is located in the WmART package's pub.art.service folder. For more information about this service, see the *webMethods Integration Server Built-In Services Reference*.

9 Click OK.

10 Click the Results tab (in Developer) or Service Result tab (in Designer) to view the output from this service.

Viewing Adapter Services

You use Developer or Designer to view adapter services.

To view an adapter service

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 In Developer or Designer, expand the package and folder that contain the service you want to view.

3 Double-click the service you want to view.

Developer or Designer displays the configured service in the service template's Adapter Service Editor.

Editing Adapter Services

You use Developer or Designer to edit adapter services.

To edit an adapter service

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 In the Developer or Designer, browse to and open the adapter service that you want to edit.

3 Double-click the service you want to edit.

Developer or Designer displays the adapter service in the Editor.

4 Do one of the following:

- If you have the VCS Integration feature enabled, right-click the service and select Check Out.
- If you do not have the VCS Integration feature enabled, right-click the service and select Lock for Edit.

5 Modify the values for the adapter service's parameters as needed. For detailed descriptions of the service's parameters, see the section on configuring a service for the specific type of service you want to edit.

6 After you have completed your modifications, save the service and do one of the following:

- If you have the VCS Integration feature enabled, right-click the service and select Check In. Enter a check-in comment and click OK.
- If you do not have the VCS Integration feature enabled, right-click the service and select Unlock.

7 Save the service.

Deleting Services

You use Developer or Designer to delete adapter services.

To delete a service

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 In the Developer or Designer, expand the package and folder that contain the service you want to delete.

3 Right-click the adapter service and click Delete.

Validating Adapter Service Values

Developer or Designer enables the Salesforce.com Adapter to validate user-defined data for adapter services at design time. You can validate the values for a single adapter service or you can configure Developer or Designer to always validate the values for adapter services. Both options could potentially slow your design-time operations.

When you enable data validation for a single adapter service, Developer or Designer compares the service values against the resource data that has already been fetched from the selected adapter.

If you select the option to always validate values for adapter services, it will do so for all webMethods WmART-based adapters installed on the Integration Server.

For more information about the Adapter Service/Notification Editor and other Developer or Designer menu options and toolbar icons, see the *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.

To enable automatic data validation for a single adapter service

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 In the Developer or Designer, expand the package and folder that contain the service for which you want to enable automatic validation.

3 Double-click the service for which you want to validate the data.

Developer or Designer displays the configured adapter service in the service template's Adapter Service Editor.

4 Click the  icon.

To always validate the values for all adapter services

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 Start Developer or Designer.

3 Perform one of the following:

- If you are using Developer, select the Tools > Options > Integration Server > Adapter Service/Notification Editor item.
- If you are using Designer, select the Window > Preferences > Software AG > Service Development > Adapter Service/Notification Editor item.

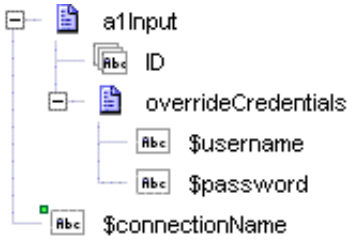
4 Enable the Automatic data validation option.

5 Click OK.

Changing the Credentials Associated with an Adapter Service at Run Time

The Salesforce.com Adapter allows users to specify an alternate connection or user credentials at run-time to invoke the adapter services.

- Configure the adapter service to use the default connection (with which the service was created) or specify an alternate connection alias, as part of the $*connectionName* input field.
- Execute the adapter service by overriding the default credentials, as part of the input signature for the adapter service. The input signature for the adapter services contains a separate document structure: 'overrideCredentials', with fields $*username* and $*password*, for specifying the username and password to be used for the adapter service execution.



Note: These dynamic credentials are valid only for the currently selected adapter service during runtime. The adapter service makes a separate login request and gets a new sessionId, with which the service call is executed. The adapter service does not use the connection from the connection pool when dynamic credentials are specified. Ensure that the credentials have sufficient privilege to perform the operation.

If there are no values specified for the $username and $password fields, the adapter service uses the default connection.

Note: When you change the credentials associated with an adapter service at runtime by overriding the default credentials, the adapter communication with Salesforce.com does not use compression.

Reloading Adapter Values

You can enable the Salesforce.com Adapter to reload and validate user-defined data for adapter services at design time in Developer or Designer. You can reload values for a single adapter service or you can configure Developer or Designer so it automatically reloads the values for adapter services. Both options could potentially slow your design-time operations.

When you reload adapter values for a single adapter service, Developer or Designer compares the service values against the resource data that has already been fetched from the selected adapter.

If you select the option to always reload values for adapter services, it will do so for all webMethods WmART-based adapters installed on the Integration Server.

For more information about the Adapter Service/Notification Editor, other menu options, and toolbar icons, see the *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.

To reload the adapter values for a single adapter service

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 In the Developer or Designer, expand the package and folder that contain the service for which you want to enable automatic validation.

3 Double-click the service for which you want to validate the data.

Developer or Designer displays the configured adapter service in the service template's Adapter Service Editor.

4 Click the  icon.

To reload the adapter values for all adapter services

1 Review the steps in "Before Configuring or Managing Adapter Services" on page 58.

2 Start Developer or Designer.

3 Perform one of the following:

- If you are using Developer, select the Tools > Options > Integration Server > Adapter Service/Notification Editor item.
- If you are using Designer, select the Window > Preferences > Software AG > Service Development > Adapter Service/Notification Editor item.

4 Enable the Automatic polling of adapter metadata option.

5 Click OK.

6 Adapter Notifications

Overview

The Salesforce.com Adapter facilitates polling of data with Salesforce.com using the adapter polling notification templates. These templates enable a user to specify the sObjects needed to poll Salesforce.com at specified periodic intervals for any changes in the organization's data. This essentially includes creating, updating, and deleting records. The adapter uses the data replication APIs provided by Force.com Web services, getUpdated() and getDeleted(), to implement the polling notification.

Before you configure the Salesforce.com Adapter notifications, you must configure the connections you plan to use with them. See "Configuring Adapter Connections" on page 48 for details.

For a description of adapter notifications, see "Adapter Notifications" on page 20.

If you are using the adapter in a clustered environment, you will need to configure the notifications so that they run effectively. For more information, see "Polling Notification Support in Integration Server Clusters" on page 41.

Before Configuring Adapter Notifications

Before you configure a Salesforce.com Adapter notification

1 Start your Integration Server and the Integration Server Administrator, if they are not already running.

2 Make sure you have Integration Server administrator privileges so that you can access the Salesforce.com Adapter's administrative screens. For information about setting user privileges, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7.

3 Using the Integration Server Administrator, make sure the WmSalesforceAdapter package is enabled. For instructions, see "Enabling and Disabling Packages" on page 36.

4 Using the Integration Server Administrator, configure an adapter connection to use with the notification. See "Configuring Adapter Connections" on page 48 for instructions.

5 Start webMethods Developer or webMethods Designer if it is not already running.

Note: If you are using Developer 6.1 or later, use the Edit perspective for all procedures unless stated otherwise. If you are using Designer, use the Service Development perspective. For more information, see the *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.

6 Using the Developer or Designer, create a user-defined package to contain the notification, if you have not already done so. When you configure notifications, you should always define them in user-defined packages. For more information about managing packages for the adapter, see "Managing the Adapter Package" on page 34.

7 You must schedule and enable a notification before you can use it.

Configuring Adapter Notifications

The polling notification template is designed to work on a single Salesforce object at a time. The object must be replicable (that is, the replicateable field for the Object must be set to "true"; for more information, see the *Force.com Web Services API Developer's Guide)*. The Salesforce.com Adapter defines two notification templates, one for capturing any create or update changes and another for any records deleted.

The Salesforce.com Adapter uses the getUpdated() call to implement Upsert Notifications and getDeleted() call for Delete Notifications. The call takes three input parameters: startDate, endDate and sObject as explained below:

- startDate: The initial start date is calculated by the notification based on the day count parameter provided by the user. For subsequent calls, the start date is used as the latestDateCovered from the previous notification.
- endDate: The current date is used as the end date by the notification.
- sObject: The object selected by the user during creation of the notification.

Creating an Upsert Notification

An Upsert Notification publishes an output document containing the IDs of the updated sObjects for the given time interval.

To configure an Upsert notification

1 Review the steps in "Before Configuring Adapter Notifications" on page 84.

2 Start Developer or Designer.

3 Perform one of the following:

- If you are using Developer, select File > New > Adapter Notification and click Next.
- If you are using Designer, perform the following:
  1 Right-click the package in which the notification should be contained and select New > Adapter Notification.
  2 Select the parent namespace and type a name for the adapter notification.
  3 Click Next.

4 Select Salesforce.com Adapter as the adapter type and click Next.

5 Select Upsert Notification as the Template Name and click Next.

6 Select the appropriate Adapter Connection Name and click Next. This specifies the connection that the notification will use to connect to the Salesforce.com resource.

7 If you are using Developer, select a package and folder to contain the notification, type a unique name for the notification, and click Next.

8 The name of the publishable document associated with this notification displays. Click Finish.

9 The editor for the adapter notification appears. The tabs that appear on the editor are explained below:

- Upsert Polling: This tab enables you to configure the notification. There are two fields on this tab.
  - DayCount: The notification uses the value in this field to calculate the initial start date when the notification runs. The value must be between 1 and 30 days.
  - Salesforce object: This field identifies the object type for which the create or update notification is required. When the notification is created, Upsert notification does the validation check to ensure that the sObject is replicable.
- Adapter Settings: Select this tab to confirm adapter service properties such as the Adapter Name, Adapter Connection Name, and Adapter Notification Template, as necessary.
- Output Fields: This tab displays the output that will be used by ART for creating the document to be published.
  - id[]: Array of IDs of each object that has been updated.
  - latestDateCovered: The timestamp of the process when the records are fetched from Salesforce within the range startdate and enddate.
- Permissions: This tab is provided by ART and contains user-specified permissions for notifications. For information about using the Permissions tab to assign an access control list (ACL) to an element, see the *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.

Make the appropriate selections on these tabs for your notification.

10 From the File menu, select Save to save your settings.

After you create a notification using Developer or Designer, you must open that notification in Integration Server Administrator to schedule and enable it. See "Managing Polling Notifications" on page 88 for information about scheduling and enabling a notification.

Creating a Delete Notification

A Delete Notification publishes an output document containing the IDs of the deleted sObjects for the given time interval.

To configure a Salesforce.com Adapter Delete notification

1 Review the steps in "Before Configuring Adapter Notifications" on page 84.

2 Start Developer or Designer.

3 Perform one of the following:

- If you are using Developer, select File > New > Adapter Notification and click Next.
- If you are using Designer, perform the following:
  1 Right-click the package in which the notification should be contained and select New > Adapter Notification.
  2 Select the parent namespace and type a name for the adapter notification.
  3 Click Next.

4 Select Salesforce.com Adapter as the adapter type and click Next.

5 Select Delete Notification as the Template Name and click Next.

6 Select the appropriate Adapter Connection Name and click Next. This specifies the connection that the notification will use to connect to the Salesforce.com resource.

7 If you are using Developer, select a package and folder to contain the notification, type a unique name for the notification, and click Next.

8 The name of the publishable document associated with this notification displays. Click Finish.

9 The editor for the adapter notification appears. The tabs that appear on the editor are:

- Delete Polling: This tab enables you to configure the notification. There are two fields on this tab.
  - DayCount: The notification uses the value in this field to calculate the initial start date when the notification runs. The value must be between 1 and 30 days.
  - Salesforce object: This field identifies the object type for which the delete notification is required. When the notification is created, Delete notification does the validation check to ensure that the sObject is replicable.
- Adapter Settings: Select this tab to confirm adapter service properties such as the Adapter Name, Adapter Connection Name, and Adapter Notification Template, as necessary.
- Output Fields: This tab displays the output that will be used by ART for creating the document to be published.

- deleteRecords[].id: The ID of the record that has been deleted.
- deleteRecords[].deleteDate: The date when the object was deleted.
- latestDateCovered: The timestamp of the process when the records are fetched from Salesforce within the range startdate and enddate.

- Permissions: This tab is provided by ART and contains user-specified permissions for notifications. For information about using the Permissions tab to assign an access control list (ACL) to an element, see the *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.

Make the appropriate selections on these tabs for your notification.

10 From the File menu, select Save to save your settings.

After you create a notification using Developer or Designer, you must open that notification in Integration Server Administrator and schedule and enable it. See "Managing Polling Notifications" on page 88 for information about scheduling and enabling a notification.

Managing Polling Notifications

You must schedule a notification and then enable it before you can use the notification. Use the Integration Server Administrator along with the following procedures to do so.

Note: You must have webMethods Integration Server administrator privileges to access the Salesforce.com Adapter's administrative screens. For information about setting user privileges, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7.

To manage polling notifications

1 In Integration Server Administrator select Adapters > Salesforce.com Adapter.

2 From the navigation area, select Polling Notifications.

3 From the Polling Notifications table, provide values for the following fields for each adapter notification:

Field	Description/Action
Notification Name	The name of the notification.
Package Name	The name of the package for the notification.

Field	Description/Action
Enabled	**Note:** You must schedule a polling notification before you can enable it. To schedule a polling notification, click the icon described in these procedures. After you schedule a polling notification, you can use this option to enable (**Yes**) or disable (**No**) a polling notification. Click on the current value in this field to change its value. Enabling and disabling a notification affects how its trigger is created and dropped. For details, see "Notification Types" on page 21. If there is no polling notification scheduled for a given adapter notification, **Not Scheduled** appears in this field. Click the icon to create a polling notification as described in step 4.
Edit Schedule	Click the icon to create or modify polling notification parameters. **Note:** You must disable a polling notification before you can edit it. Continue to step 4.
View Schedule	Click on the **View Schedule** icon to review the parameters for the selected polling notification. Click **Return to Salesforce.com Adapter Notifications** to go back to the main polling notification page.

4 To create or modify schedule parameters for the selected adapter notification, click the icon and provide values for the following fields:

Field	Description/Action
Interval (seconds)	Type the polling interval time in seconds.
Overlap	**Note:** Do not use this option; otherwise, when you enable this notification, it may lock up tables and cause the Integration Server to fail.
Immediate	Enable this option to start polling immediately.

5 Click **Save Schedule**.

6 After you create a polling notification, you can enable it. Use the **Enabled** field described in step 3 to enable a polling notification.

Exporting Configured Adapter Notifications

You can export notifications from one Integration Server to another Integration Server. You do not need to disable notifications in order to export them. In most cases, the current state of the notifications in the package that you export is retained. However, if you deploy to a different Integration Server and connect to a different database, then you should first disable the notification.

Note: A given notification can run on only one Integration Server at a time.

The start date of the notification is stored in the shared cache location. When the notifications are exported, if the notification is unable to fetch the start date from the cache, the start date for the first notification will be calculated based on the day count configured for the adapter notification.When the Integration Server with the exported notifications starts, each configured notification starts to poll the changed data from Salesforce.com.

If you want to export configured notifications in a disabled state, you need to disable the notifications before you export the package containing them. To enable or disable a notification, see "Managing Polling Notifications" on page 88.

For more information about Insert Notifications, Update Notifications, or Delete Notifications, see "Notification Types" on page 21.

For more information about exporting packages, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7.

Viewing Notifications

You use the Developer or Designer to view notifications.

To view a notification

1 Review the steps in "Before Configuring Adapter Notifications" on page 84.

2 In Developer or Designer, expand the package and folder that contain the notification you want to view.

3 Double-click the notification you want to view.

Developer or Designer displays the notification in the notification template's Adapter Notification Editor.

Editing Notifications

You use the Developer or Designer to edit notifications.

To edit a notification

1 Review the steps in "Before Configuring Adapter Notifications" on page 84.

2 In Developer or Designer, expand the package and folder that contain the notification you want to edit.

3 Select the notification you want to edit.

Developer or Designer displays the notification in the notification template's Adapter Notification Editor.

4 Modify the values for notification's parameters as needed. For detailed descriptions of the notification's parameters, see "Configuring Adapter Notifications" on page 85 for the specific type of notification you want to edit.

Note: When you edit a notification, all the parameters in the adapter notification are reset.

Deleting Notifications

You use the Developer or Designer to delete adapter notifications.

Note: Before you delete the notification, be sure that you first disable it. To disable a notification, see "Managing Polling Notifications" on page 88.

To delete a notification

1 Review the steps in "Before Configuring Adapter Notifications" on page 84.

2 In the Developer or Designer, expand the package and folder that contain the notification you want to delete.

3 Right-click the notification and click **Delete**.

Validating Adapter Notification Values

Developer and Designer enable the Salesforce.com Adapter to validate user-defined data for adapter notifications at design time. You can validate the values for a single notification or you can configure the Developer or Designer to always validate the values for notifications. Both options could potentially slow your design-time operations.

When you enable data validation for a single adapter notification, Developer or Designer compares the notification values against the resource data that has already been fetched from the selected adapter.

If you select the option to always validate values for adapter notifications, it will do so for all webMethods WmART-based adapters installed on the Integration Server.

For more information about the Adapter Service/Notification Editor, other Developer or Designer menu options, and toolbar icons, see the *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.

To validate adapter notification values for a single adapter service

1 Review the steps in "Before Configuring Adapter Notifications" on page 84.

2 In the Developer or Designer, expand the package and folder that contain the notification for which you want to enable automatic validation.

3 Double-click the notification for which you want to validate the data.

Developer or Designer displays the configured adapter notification in the service template's Adapter Notification Editor.

4 Click the  icon.

To validate adapter notification values for all notifications

1 Review the steps in "Before Configuring Adapter Notifications" on page 84.

2 Start Developer or Designer.

3 Perform one of the following:

- If you are using Developer, select the Tools > Options > Integration Server > Adapter Service/Notification Editor item.
- If you are using Designer, select the Window > Preferences > Software AG > Service Development > Adapter Service/Notification Editor item.

4 Enable the Automatic data validation option.

5 Click OK.

Reloading Adapter Values

Developer and Designer enable the Salesforce.com Adapter to reload and validate user-defined data for notifications at design time. You can reload values for a single notification or you can configure Developer or Designer to always validate the values for notifications. Both options could potentially slow your design-time operations.

When you reload adapter values for a single adapter notification, Developer or Designer compares the notification values against the resource data that has already been fetched from the selected adapter.

If you select the option to always reload values for adapter notifications, it will do so for all webMethods WmART-based adapters installed on the Integration Server.

For more information about the Adapter Service/Notification Editor, other menu options, and toolbar icons, see the *webMethods Designer Online Help* or the appropriate Developer user's guide for your release in "Document Titles" on page 7.

To reload the adapter values for a single adapter notification

1 Review the steps in "Before Configuring Adapter Notifications" on page 84.

2 In the Developer or Designer, expand the package and folder that contain the service for which you want to enable automatic validation.

3 Double-click the service for which you want to validate the data.

Developer or Designer displays the configured adapter service in the service template's Adapter Notification Editor.

4 Click the  icon.

To automatically reload the values for all adapter notifications

1 Review the steps in "Before Configuring Adapter Notifications" on page 84.

2 Start Developer or Designer.

3 Perform one of the following:

- If you are using Developer, select the **Tools** > **Options** > **Integration Server** > **Adapter Service/Notification Editor** item.
- If you are using Designer, select the **Window** > **Preferences** > **Software AG** > **Service Development** > **Adapter Service/Notification Editor** item.

4 Enable the **Automatic polling of adapter metadata** option.

5 Click **OK**.

7 Support for Salesforce.com Outbound Messaging

Overview

Outbound messaging is part of the workflow rule functionality in Salesforce.com. Workflow rules watch for specified field changes and trigger automatic sending of an outbound message. Using the notifications() call in Salesforce.com, outbound messaging sends SOAP messages over HTTP or HTTPS to a designated endpoint when triggered by a workflow rule.

You can configure workflow rules in Salesforce.com so that it sends notifications when any changes are made to Salesforce objects. Once a workflow rule has been associated with an outbound message, a WSDL is created and is made available for download from the Salesforce.com user interface. This WSDL contains the endpoint URL for the recipient of the message and the information you want to include in the outbound message.

If you configure an outbound message, then whenever an event defined in the workflow rule is triggered, Salesforce.com sends an outbound message containing the values of the specified fields to your Integration Server. You create a listener in the Integration Server to process the outbound notification message resulting from the workflow rule. For more information, see "Support for Receiving Outbound Messages from Salesforce.com" on page 23.

To ensure security, place your Integration Server behind an internal firewall and place a Reverse HTTP Gateway in the DMZ to allow it to accept communications and process requests from Salesforce.com. For information related to setting up a Reverse HTTP Gateway Server for processing Salesforce.com notifications, see "Setting Up a Reverse HTTP Gateway Server" on page 99. For detailed information about using a Reverse HTTP Gateway Server, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7.

Note: The Reverse HTTP Gateway Server is referred to as the Reverse Invoke Integration Server in Integration Server 6.5.

Setting up to Receive the Outbound Messages from Salesforce.com

This section lists the steps you perform to set up Salesforce.com to send outbound messages and the Integration Server to receive the outbound message notifications from Salesforce.com.

To set up Integration Server and Salesforce.com for outbound messaging

1 Define the outbound message and configure the workflow rule using the Salesforce.com user interface. When you define an outbound message, Salesforce.com creates a WSDL file that is associated with the outbound message. For more information about the steps to define the outbound message, see the *Force.com Web Services API Developer's Guide*.

2 Download the WSDL file using the Salesforce.com user interface. Use this WSDL file when creating the listener for the outbound message. For more information, see "Downloading the WSDL for the Outbound Message" on page 97.

3 Create a listener to handle the outbound notification message resulting from the workflow rule. The procedure to implement the listener depends on the version of the Integration Server. For more information, see "Creating a Listener in Integration Server for Receiving the Outbound Message" on page 98.

4 Set up a Reverse HTTP Gateway Server for receiving the outbound messages from Salesforce.com. For information, see "Setting Up a Reverse HTTP Gateway Server" on page 99.

5 Update the endpoint URL of the Salesforce.com outbound message. Specify the gateway external port of the Reverse HTTP Gateway Server as the endpoint URL for the outbound message. For more information, see "Updating the Endpoint URL of the Salesforce.com Outbound Message" on page 101.

Downloading the WSDL for the Outbound Message

After defining the outbound message and the workflow rules in Salesforce.com, Salesforce.com creates a WSDL that is associated with the message. This section lists the steps to download the WSDL file using the Salesforce.com user interface. Use this WSDL file when creating the listener for the outbound message.

To obtain the WSDL for the outbound message

1 In the Salesforce.com user interface, go to Setup > Create > Workflow & Approvals > Outbound Messages.

2 On the All Outbound Messages page, click the name of the message for which you want to create a service.

3 In the Endpoint WSDL field, click Click for WSDL to display the WSDL.

4 Save the WSDL to your file system. You will use this WSDL when creating the listener that handles the outbound messages.

For more information on how to download the WSDL file, see the *Force.com Web Services API Developer's Guide*.

Creating a Listener in Integration Server for Receiving the Outbound Message

In Integration Server, you can create a listener for the outbound message from Salesforce.com. The procedure for creating a listener in Integration Server 7.1.2 and later is different from the procedure in Integration Server 6.5 SP3.

To create the listener that processes the outbound message in Integration Server 7.1.2 and later

1 Create a Provider Web service descriptor using the WSDL file from Salesforce.com.

   a In Developer or Designer, select File > New.

   b Select Web Service Descriptor and click Next.

   c On the Web Description Window, select the following and click Next.

   - For Create Web service descriptor as, select Provider.
   - For Web service source, select WSDL URL. If you are using Designer, click Browse to select the WSDL source.
   - For Enforce WS-I Basic Profile 1.1 compliance, select Yes or No. The default is No.

   d Provide a name for the service, select where to save the service, and click Next.

   e If you are using Developer, browse to the WSDL source.

   f Click Finish.

   Developer or Designer creates the Provider Web service descriptor and a flow service with the name, notifications. The input and output signatures of this flow service will be defined.

2 Code the logic in the empty notifications flow service. You can add any custom code for further processing.

   Because Salesforce.com expects an acknowledgment to the outbound message that it sends, the business logic should also include steps to map the output signature with the required value.

To create the listener that processes the outbound message in Integration Server 6.5 SP3

1 Import the WSDL.

   Using Developer, create a Web service connector from the WSDL that you downloaded from Salesforce.com. Developer validates the WSDL you selected, creates the connector and supporting Integration Server elements, and saves the connector and the elements to the folder you specified.

   For more information on how to create a Web service connector, see the webMethods *Web Services Developer's Guide 6.5*.

2 Create an empty flow service. For information on how to create a flow service, see *webMethods Developer User's Guide 6.5*.

3 Specify the input and output signatures for the service.

The service input signature must contain a *soapRequestData* object and a *soapResponseData* object, and the output signature must contain a *soapResponseData* object. You can use pub.soap.utils:requestResponseSpec specification to specify the inputs and outputs for the service.

The *soapRequestData* object will contain the outbound message posted by Salesforce.com. The *soapResponseData* object will contain the response that Salesforce.com expects as an acknowledgement to the outbound message that it posted.

The IData representation of these input and output signatures can be found in the Document Types created in step 1.

4 Based on your business use, code the logic that the listener should execute on receiving the outbound message.

The logic should also include steps to send back a SOAP Response as an acknowledgement.

Note: The response message must include the XML elements that correspond to the output Document Type generated in step 1. Map the output SOAP message to *soapResponseData* object.

---

5 Set the Universal Name property for the flow service.

a Set the value of the Namespace name field to the value of *targetNamespace* defined in the WSDL.

b Set the value of Local name to "notifications" as described in the WSDL. A part of the WSDL document mentioning the local name is given below:

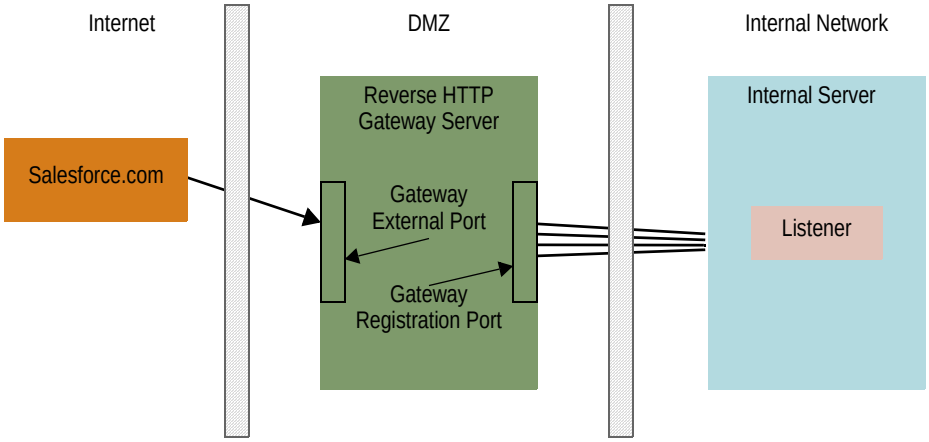
```
<message name="notificationsRequest">
<part element="tns:notifications" name="request"/>
</message>
```

For more information about specifying an explicit universal name, see the *webMethods Developer User's Guide*.

Setting Up a Reverse HTTP Gateway Server

This section provides the basic steps for setting up a Reverse HTTP Gateway Server. For detailed information, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7.

Note: The Reverse HTTP Gateway Server is referred to as the Reverse Invoke Integration Server in Integration Server 6.5.



To set up a Reverse HTTP Gateway Server

1 Install an Integration Server in your DMZ to be your Reverse HTTP Gateway Server.

2 Disable the Integration Server "Developer" and "Replicator" users because you will not need these users on a Reverse HTTP Gateway Server. Disabling these users prevents someone from gaining access to your Reverse HTTP Gateway Server through them.

3 Set up the gateway external port (or the proxy port in case of Integration Server 6.5) to make the Integration Server a Reverse HTTP Gateway Server. The gateway external port is the port through which the Reverse HTTP Gateway Server will listen for the requests from Salesforce.com.

   The endpoint URL of the Salesforce.com outbound message must point to the gateway external port of the Reverse HTTP Gateway Server. For more information, see "Updating the Endpoint URL of the Salesforce.com Outbound Message" on page 101.

4 Set up the gateway registration port. Through this port the Reverse HTTP Gateway Server maintains its connection to the Internal Server and routes the outbound message to the listener.

   Create the connections from the Internal Server to the Reverse HTTP Gateway Server (or Reverse Invoke Integration Server). For more information, see the appropriate Integration Server administration guide for your release in "Document Titles" on page 7.

Updating the Endpoint URL of the Salesforce.com Outbound Message

After setting up a Reverse HTTP Gateway Server, you must update the endpoint URL specified in the outbound message to point to the external port of the Reverse Invoke Integration Server. The procedure for updating the endpoint URL in Integration Server 7.1.2 and later is different from the procedure in Integration Server 6.5 SP3.

Note: The Reverse HTTP Gateway Server is referred to as the Reverse Invoke Integration Server in Integration Server 6.5.

To update the endpoint URL of the outbound message for Integration Server 7.1.2 and later

1 In the Salesforce.com user interface, go to **Setup > Create > Workflow & Approvals > Outbound Messages**.

2 Click **Edit** to make changes to an existing outbound message.

3 View the WSDL document on which the listener is based. For more information about how to acquire a WSDL document for the Provider Web Service Descriptor, see the webMethods *Web Services Developer's Guide*.

The value of the `address location` element in the WSDL document currently specifies the location of the listener.

4 Use the value of the `address location` element in the WSDL and replace the *<IP>:<port>* part of this address with *<Reverse HTTP Gateway Server IP>:<Gateway External Port>* to create an endpoint URL for the outbound message.

When Salesforce.com sends an outbound message to this newly created endpoint URL, the outbound message is routed through the Reverse HTTP Gateway Server and reaches the listener service in the Internal Server.

For more information about creating the listener for the outbound message from Salesforce.com, see "To create the listener that processes the outbound message in Integration Server 7.1.2 and later" on page 98.

5 In the **Endpoint URL** field, enter the endpoint URL for the outbound message (that is, the endpoint URL you created in step 4).

For example, let us consider that the value of the `address location` element in the WSDL is `http://10.60.28.176:5555/ws/OutboundMessaging:testWSDLDescriptor`, the IP address of the Reverse HTTP Gateway Server is `10.30.20.110`, and the Gateway External Port is `7777`.

If you replace `10.60.28.176:5555` with `10.30.20.110:7777` in the address, the resulting address will be `http://10.30.20.110:7777/ws/OutboundMessaging:testWSDLDescriptor`

Then, you will specify `http://10.30.20.110:7777/ws/OutboundMessaging:testWSDLDescriptor` as the endpoint URL for the outbound message.

6 Click **Save**.

To update the endpoint URL of the outbound message for Integration Server 6.5 SP3

1 In the Salesforce.com user interface, go to **Setup > Create > Workflow & Approvals > Outbound Messages**.

2 Click **Edit** to make changes to an existing outbound message.

3 In the **Endpoint URL** field, enter the endpoint URL of the proxy port of the Reverse Invoke Integration Server in the following format:

http://<*Reverse Invoke Integration Server IP*>:<*proxy port*>/soap

For more information about creating the listener for the outbound message from Salesforce.com in Integration Server 6.5 SP3, see "To create the listener that processes the outbound message in Integration Server 6.5 SP3" on page 98.

4 Click **Save**.

For more information about configuring the outbound messages, see the *Force.com Web Services API Developer's Guide*.

8 Logging and Exception Handling

Overview

The following sections describe Salesforce.com Adapter message logging, exception handling, and error codes. A list of error codes and supporting information appears at the end of this chapter.

Adapter Logging Levels

The Salesforce.com Adapter uses the Integration Server logging mechanism to log messages. You can configure and view the Integration Server logs to monitor and troubleshoot the Salesforce.com Adapter.

For more information about logging, in the Integration Server, including instructions for configuring and viewing the different kinds of logs supported by the server, see the appropriate logging guide for your release in "Document Titles" on page 7.

Configuring Adapter Logging Levels

If you are using Integration Server 7.1.2 or later, you can configure different logging levels for the Salesforce.com Adapter.

To access the adapter's logging information and change logging settings

1 From the Integration Server Administrator, select Settings > Logging.

   On the Logging Settings screen, the Loggers section has Adapters included in the Facility section.

2 Expand the Adapters tree to see a list of all installed adapters with their code number and adapter description, along with the logging level.

3 Select Edit Logging Settings.

4 Select the required Level of logging for the Salesforce.com Adapter.

5 After making your changes, click Save Changes.

For complete information about specifying the amount and type of information to include in the log, see the appropriate logging guide for your release in "Document Titles" on page 7.

Salesforce.com Adapter Message Logging

The Integration Server maintains several types of logs. However, the Salesforce.com Adapter logs messages only to the audit, error, and server logs. Because the Salesforce.com Adapter works in conjunction with the WmART package, the adapter's messages and exceptions typically appear within log messages for the WmART package.

The logging levels are different depending on which version of the Integration Server you are running the adapter on, as shown in the following table:

Integration Server	Log	Description
Integration Server 7.1.2 or later	Audit Log	You can monitor individual adapter services using the audit log as you would audit any service in the Integration Server. The audit properties for an adapter service are available in the Salesforce.com Adapter service template in the Properties panel.
	Error Log	The Salesforce.com Adapter automatically posts fatal-level and error-level log messages to the error log. These log messages will appear as Adapter Runtime messages.
	Server Log	The Salesforce.com Adapter posts messages to the server log, depending on how the server log is configured. Fatal-level through debug-level log messages appear as Adapter Runtime log messages. Trace-level log messages appear as Salesforce.com Adapter log messages.
Integration Server 6.5	Audit Log	You can monitor individual adapter services using the audit log as you would audit any service in the Integration Server. The audit properties for an adapter service are available in each Salesforce.com Adapter service template in the Properties panel.
	Error Log	The Salesforce.com Adapter automatically posts critical-level and error-level log messages to the Integration Server's Error log. These log messages will appear as Adapter Runtime messages.
	Server Log	The Salesforce.com Adapter posts messages to the Integration Server log, depending on how the server log is configured. Critical-level through debug-level log messages appear as Adapter Runtime log messages. V1-Verbose1 or V4-Verbose4 log messages appear as Salesforce.com Adapter log messages.

The Salesforce.com Adapter's log messages appear in the following format: ADA.0505.*nnnnc*, where:

- ADA is the facility code that indicates that the message is from an adapter.
- 0505 is the Salesforce.com Adapter major code, which indicates that the error is generated by the Salesforce.com Adapter.
- *nnnn* represents the error's minor code.
- *c* represents the message's severity level (optional).

To monitor the Salesforce.com Adapter log messages in the Server log, ensure that your server log's logging settings are configured to monitor the following facilities:

- 0113 Adapter Runtime (Managed Object)
- 0114 Adapter Runtime
- 0117 Adapter Runtime (Adapter Service)
- 0118 Adapter Runtime (Connection)
- 0126 Adapter Runtime (System Contract Component [SCC] Connection Manager)

Salesforce.com Adapter Exception Handling

Exceptions raised by the adapter will contain the SOAP fault errors that Salesforce.com returns in its error stack.

The Force.com Web services API calls provides the following types of error handling:

- For errors resulting from badly formed messages, failed authentication, or similar problems, the Force.com Web services API returns a SOAP fault message with an associated Exception Code.
- For most calls, if the error occurs because of a problem specific to the query, the API returns an Error. For example, if a create request contains more than 200 objects, the API returns an Error.

For more information about Salesforce.com error handling, see the *Force.com Web Services API Developer's Guide*.

When creating a flow or Java service that incorporates an adapter service, you might want to build logic into the wrapping service to catch and handle exceptions.

Adapter Error Codes

This section lists the Salesforce.com Adapter error codes and provides information on the message, reason, and possible action for each error.

1001 Resource Connection Exception.

Cause: This message is displayed either when there is an error while connecting to Salesforce.com or when the login request encounters an API fault with INVALID_SESSION_ID for the current user.

Response: No user intervention required. The adapter will automatically try to reset the connection pool.

1002 Register Resource Domain Exception.

Cause: There is an error while trying to register the resource domains for the adapter. This exception is due to the occurrence of an error either while trying to create a new adapter service or while loading the metadata information for any existing adapter service.

Response: If the exception occurs even after a retry, contact Software AG Global Support.

1005 Unable to describe global types. *exception_message*

Cause: The adapter is unable to get the list of available Salesforce objects.

Response: Ensure that the connection parameters are valid, and then retry the operation.

1006 Unable to describe *sObject_list. exception_message*

Cause: The adapter is unable to describe one or more Salesforce objects requested in the describeSObjects() call.

Response: Ensure that the connection parameters are valid, and then retry the operation.

1007 Please be sure that you have a valid username and password.

Cause: Incorrect username or password specified.

Response: The user should either use a different set of login credentials or use the correct credentials.

1010 Your password has expired.

Cause: An expired password was used to log in to Salesforce.com. This error can also occur when the adapter tries to log in again to keep the session alive, but the password has expired.

Response: Specify a different set of login credentials or get the password renewed by Salesforce.com.

1011 API Fault occurred while executing the call [*API_call_name*], Code: *exception_code*, Message: *exception_message*

Cause: There is a fault in the indicated Force.com API Web service call. The exception code and message thrown by the call are reported.

Response: Retry the call after making the appropriate changes as specified by the API.

1012 Unknown utility service: *service_name*

Cause: This error occurs while trying to execute a utility service that is not implemented.

This error indicates that an unknown utility API is trying to execute via the indicated Utility Operation adapter service.

Response: Contact Software AG Global Support, and report the error.

1013 Error occurred while trying to get the resource domain values for service: *service_name* and resource domain:*domain_name*

Cause: Resource domain lookup failed for the indicated adapter service and resource domain.

Response: Ensure that the connection parameters are valid, and then retry the operation.

1015 No output type description available for service *service_name*.

Cause: Unknown or unhandled type description encountered for the indicated adapter service.

Response: Contact Software AG Global Support, and report the error.

1016 Service Exception thrown while executing the call [*adapter_service_operation*].

Cause: Service exception thrown while executing the adapter service.

Response: Check the error logs for more information, and take the necessary action.

1017 Remote Exception occurred during the execution of [*adapter_service_operation*].

Cause: Remote exception thrown when an error occurs while the indicated adapter service is communicating with Salesforce.com.

Response: Check the error logs for more information, and take the necessary action.

1018 Mandatory input missing.

Cause: This error occurs when you do not specify the required input for the adapter service.

Response: Rerun the adapter service and specify the required input.

1019 Unable to map the value [*value*] to type [*data_type*]. Message: *exception_message*

Cause: This error occurs when the adapter tries to map the indicated value to a supported data type. The specific exception message is reported.

Response: Contact Software AG Global Support, and report the error.

1020 Invalid Argument passed for the field [*input_field*] while invoking [*web_service*] service.

Cause: This error occurs when an invalid value is specified for the indicated input field while invoking a Web service.

Response: Specify a correct value to the input field, and then retry the operation.

1021 The library files required for enabling the compression functionality are missing. For more information, see the webMethods Salesforce.com Adapter Installation and User's Guide.

Cause: The jar files required by the adapter to enable compression in the SOAP request and response messages are missing.

Response: See the instructions in "Configuring Adapter Connections" on page 48.

1050 The day count must be an integer between 1 and 30.

Cause: This error occurs if you specify an incorrect value for the DayCount field while configuring an Upsert or Delete notification.

Response: Specify a value between 1 and 30.

1051 The selected sObject is not replicable.

Cause: This error occurs if you selected a non-replicable Salesforce object while configuring an Upsert or Delete notification.

Response: Select the correct Salesforce object while configuring an Upsert or Delete notification. For more information on replicable objects, see the *Force.com Web Services API Developer's Guide*.

1052 An sObject is mandatory.

Cause: This error occurs while validating a notification that does not specify which Salesforce object is to be monitored.

Response: Specify the Salesforce object to be monitored by the notification.

1053 Unable to retrieve the start date for the notification.

Cause: This error occurs while retrieving the start date from the persistence store. The error can occur either due to a network issue or by corrupted data in the persistence store. Depending on the version of Integration Server, the persistence store could be the repository or the shared cache.

Response: Restart the notification after correcting the issue.

1059 Unable to invoke pub.storage service *service_name*.

Cause: This error occurs when the adapter tries to invoke the indicated pub.storage service to save or retrieve data from the persistence store.

Response: Check the error logs for more details.