webMethods Siebel Adapter Installation and User's Guide

Version 6.0 SP3

July 2012

WEBMETHODS

Document ID: ADAPTER-SBL-IUG-60 SP3-20210331

Table of Contents

About this Guide

This guide describes how to configure and use the webMethods Siebel Adapter Version 6.0 Service Pack 3. It contains information for administrators and application developers who want to exchange data with Siebel applications.

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

- The operation of Siebel Enterprise applications, and the definition and use of Siebel objects

  For information about Siebel objects, see the *Tools Guides* on the Siebel Bookshelf.

- Terminology and basic operations of your operating system
- The basic concepts and tasks of webMethods Integration Server and Digital Event Services or Software AG Designer

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format *folder.subfolder.service*, APIs, Java classes, methods, properties.
Italic	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	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 (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at http://documentation.softwareag.com.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at https://empower.softwareag.com/.

You can find product information on the Software AG Empower Product Support website at https://empower.softwareag.com.

To submit feature/enhancement requests, get information about product availability, and download products, go to Products.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the Knowledge Center.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.aspx and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at http://techcommunity.softwareag.com. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 Overview of the Adapter

About the Siebel Adapter

The webMethods Siebel Adapter is an add-on to the webMethods Integration Server that enables you to exchange data with Siebel applications. The adapter provides seamless, real-time communication to and from Siebel applications without requiring changes to your existing application infrastructure.

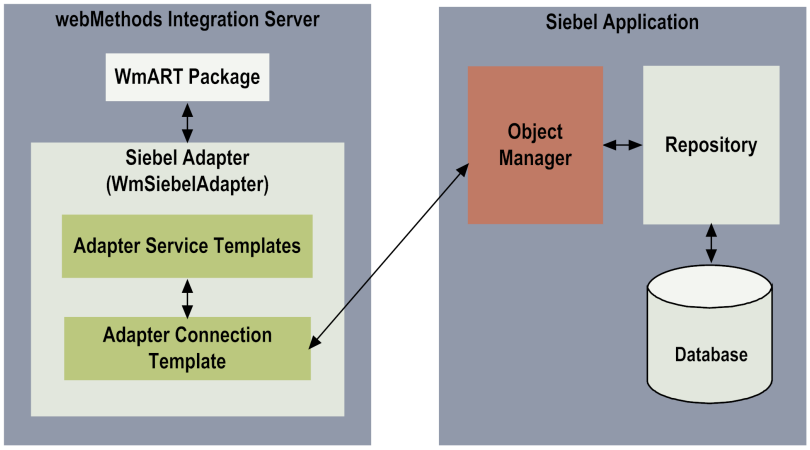
Using the Siebel Adapter, Integration Server clients can run adapter services that retrieve data from, write data to, and invoke methods in Siebel applications. Clients can also interact with webMethods from a Siebel application. To do this, they can use the webMethods COM interface or custom Siebel business service methods known as EAI webMethods Transports.

For example, you might use the Siebel Adapter to receive new information about a sales contact from an Integration Server client (IS client), and then update a Contact business component record in the Siebel application.

Architectural Overview

The Siebel Adapter provides a set of user interfaces, services, and templates that enable you to create integrations with Siebel applications. The adapter is provided as a single package that must be installed on the Integration Server. For detailed installation instructions and software requirements, see "Installing, Upgrading, and Uninstalling the Siebel Adapter " on page 37.

The following diagram shows at a high level, how an adapter service uses an adapter connection to connect to and perform an operation on your Siebel system. For more detailed information about these components, refer to the text following the diagram.

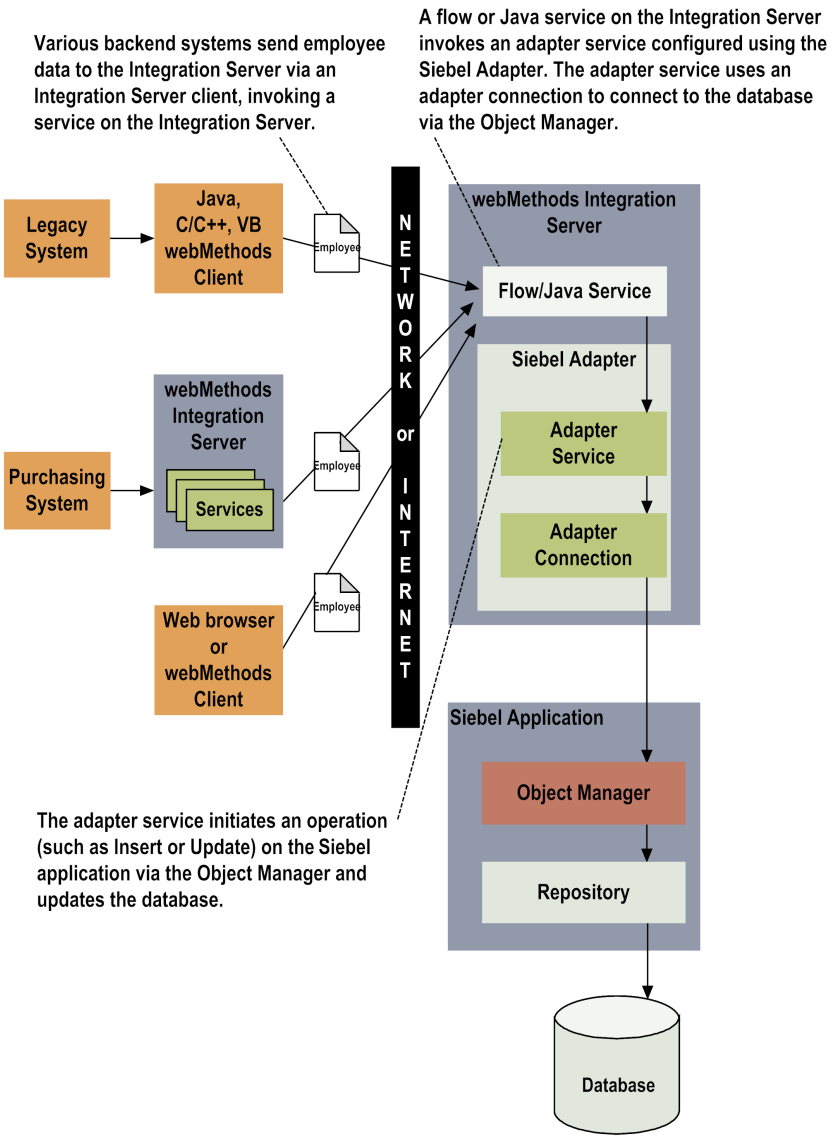


- **webMethods Integration Server:** The Siebel Adapter is installed and runs on the Integration Server.
- **WmART Package:** The WmART package provides a common framework for webMethods adapters (version 6.0 and later) to use the Integration Server functionality, making the Integration Server the run-time environment for the Siebel Adapter. The WmART package is installed with the Integration Server.
- **Siebel Adapter:** The Siebel Adapter is delivered as a single package called WmSiebelAdapter.

The Siebel Adapter provides Integration Server Administrator user interfaces that enable you to configure and manage adapter connections, and Software AG Designer user interfaces that enable you to configure and manage adapter services.

- **Adapter Connection Templates:** Adapter connection templates, which are provided with the Siebel Adapter, enable you to configure connections that the adapter uses to connect to a Siebel Server. You must configure a connection before you can configure adapter services. For details, see "Adapter Connections" on page 13.
- **Adapter Services Templates:** Adapter service templates, which are provided with the Siebel Adapter, enable you to configure adapter services that Integration Server uses to initiate and perform operations (such as Query, Insert, Update, Delete, and others) on Siebel applications. For example, an adapter service could query your Siebel application to determine whether an account contact exists in your Siebel system. For details, see " webMethods-to-Siebel Communication" on page 18.
- **Siebel Application:** The Object Manager (also known as the Business Objects Layer) resides in the Siebel application and connects to one or more repositories, each of which defines its own set of business objects and business components.

The following diagram shows a business integration where an adapter service is used to update a business component with employee data. The employee data could be provided by several different types of external Integration Server clients.



Package Management

The Siebel Adapter is provided as a package called WmSiebelAdapter, which you manage like any package on the Integration Server. In addition, you create user-defined packages for your connections and adapter services.

There are several considerations regarding how you set up and effectively manage your packages on the Integration Server:

- Understand how package dependencies work so you make the best decisions regarding how you manage your adapter services
- Enable and disable packages
- Control which development groups have access to which adapter services

- Understand how clustering, an advanced feature of the Integration Server, works to effectively manage your adapter services

For details, see "Package Management" on page 51.

Adapter Connections

The Siebel Adapter connects to a Siebel Server at run time. You configure one or more connections at design time to use in integrations. The number of connections you configure, and the types of those connections, depend on your integration needs. For example, if you have multiple Siebel installations, you can access each one using different connections.

Siebel Adapter connections contain Siebel connection parameters that:

- Enable the adapter connection to log in to a Siebel application.
- Define the type of Siebel client that the adapter uses to connect to the Siebel Server (see "Connection Types" on page 13).
- Enable the Integration Server to manage a pool of connection objects at run time (see "Connection Pooling" on page 14).
- Enable the adapter connection to use the load balancing available with Siebel Server 7.7 and later. For more information, see "Using Connections with Siebel Server Load Balancing" on page 14.

You configure connections using the Integration Server Administrator. You must have webMethods administrator privileges to access the Siebel Adapter's administrative screens.

For instructions to configure and manage Siebel Adapter connections, see " Siebel Adapter Connections" on page 59. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide* for your release.

For a list of tasks that you must complete before you can configure your connections, see "Before Configuring or Managing Adapter Connections" on page 60.

Connection Types

There are two ways in which the Siebel Adapter can connect to your Siebel Server:

- **Siebel Java Connection**: This type of adapter connection uses Siebel's Java Data Bean Business Object Interface (BOI) API to communicate with the Siebel Server. This connection requires that your Windows or UNIX platform supports a JVM that the Data Bean supports. For information about the supported versions, see "Installing, Upgrading, and Uninstalling the Siebel Adapter " on page 37.
- **Siebel Windows Thin Connection**: This type of adapter connection uses Siebel's COM Data Control client to communicate with the Siebel Server's Object Manager.

Both connection types use .srf files used by the Siebel Server.

Connection Pooling

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. All adapter services use connection pooling.

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.

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 connection's **Minimum Pool Size** field. 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 the **Pool Increment Size** field) and adds them to the connection pool. If the pool is full (as specified in the **Maximum Pool Size** field), the requesting service will wait for the Integration Server to obtain a connection, up to the length of time specified in the **Block Timeout** field, 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 the **Expire Timeout** field.

If the connection pool initialization fails because of a network connection failure or some other type of exception, you can enable the system to retry the initialization any number of times, at specified intervals.

For more information about connection pools, see "Run-Time Behavior of Connection Pools in Production Environments" on page 60. For information about configuring connections, see " Siebel Adapter Connections" on page 59.

Using Connections with Siebel Server Load Balancing

Load balancing is a new feature introduced in Siebel Server 7.7. With this feature, depending on the number of concurrent client sessions, multiple Siebel servers can share the system work load. To make this load balancing possible, Siebel introduced the concept of a *virtual server*, which is a logical grouping of several Siebel servers.

The information about the underlying physical Siebel servers is abstracted from the Siebel clients. When the client requests a connection to a virtual server, the Siebel Adapter provides it. The Siebel Java Data Bean, through which a Siebel Java Connection connects to the Siebel server, needs to know the mapping between the virtual server and the underlying Siebel servers so that it can connect to the least loaded Siebel server and start the session. This mapping is provided through a configuration file required by the Siebel Java Data Bean, called siebel.properties, which must reside in the *Integration Server_directory* \packages\WmSiebelAdapter\resources directory. In this file you specify the virtual server and then provide the list of servers used in the workload for a particular Siebel application.

Then, when configuring an adapter connection for a **Siebel Java Connection**, you provide the virtual server name in place of the gateway server and leave the **Siebel Server** field empty.

For example, say there are two Siebel servers, one running on the host `orpheus:2321` and the second on the host `eurydice:2321`. Both servers are logically grouped with the virtual server `callcenter_srv`. When the Siebel client requests a connection to the Siebel system named `callcenter_srv`, the Siebel Adapter reads the siebel.properties file to look for a virtual server entry named `callcenter_srv`, finds the list of physical servers represented by `callcenter_srv`, and connects to either `orpheus:2321` or `eurydice:2321`.

To use Siebel Server load balancing with the adapter

1. Create the siebel.properties file:

   a. Using a text editor, create a file named siebel.properties in the *Integration Server_directory* \packages\WmSiebelAdapter\resources directory.

   b. Edit the file and add the following property:

   ```
   siebel.conmgr.virtualhosts=virtualserver=1:server1:port,2:server2:port,..
   .N:serverN:port;
   ```

   where:

   - *virtualserver* is the name of the Siebel virtual server
   - *server1*, *server2*,...*serverN* are the names of the Siebel load balancing servers that are available to the adapter connection
   - *port* is the port number of the associated Siebel load balancing server

     For example, to use the servers named `orpheus` and `eurydice` (both on port 2321) that are grouped with the virtual server named `callcenter_srv`, to handle the processing for the call center, use the following property:

     ```
     siebel.conmgr.virtualhosts=callcenter_srv=1:orpheus:2321,2:eurydice:2321;
     ```

     You can also specify multiple Siebel virtual servers as needed for your configuration. For example, in addition to the call center server shown above, you also want to use the servers named `corvette`, `firebird`, and `gto` that are grouped with the virtual server named `sales_srv` to handle the processing for sales requests. In this case, your property would look like this:

     ```
     siebel.conmgr.virtualhosts=callcenter_srv=1:orpheus:2321,2:eurydice:2321;

     sales_srv=1:corvette:2321,2:firebird:2321,3:gto:2321;
     ```

   c. Save and close the file.

2. Restart the Integration Server.

> **Note:**
> Restarting the Integration Server causes the siebel.properties file to be added to the classpath, allowing the Siebel Java Data Bean to locate the file when you enable the adapter connection.

3. When configuring the adapter connections, ensure the following connection parameters are set as follows:

   - Select the Connection Type as **Siebel Java Connection**.
   - Set the **Gateway Server** parameter to be the Siebel virtual server name.
   - Leave the **Siebel Server** parameter blank.

   For more information about these connection parameters, see "Configuring Adapter Connections" on page 61.

Built-In Services For Connections

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 and 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.

Another built-in service, pub.art.service:setAdapterServiceNodeConnection, enables you to change the connection associated with an adapter service. For more information, see "Changing the Connection Associated With an Adapter Service at Design Time" on page 19.

For more information about available services, see the *webMethods Integration Server Built-In Services Reference* for your release.

Siebel Business Objects and Business Components

Siebel applications define a data abstraction layer that removes dependencies on the underlying databases. To access a database, Siebel applications must interact with the Siebel Object Manager. The Object Manager presents database table data in the form of business component records that represent database structures.

A *business component* typically represents a table in a database. Business components contain records and fields similar to those in relational database tables. For example, the Contact business object might contain the business components Customer Contact, Supplier Contact, and so on. The Customer Contact business component might contain such fields as Name and Last Contact Date.

A *business object* is a group of related business components. Each business component can have relationships with other business components. A relationship can be a parent/child relationship that has a one-to-many relationship (a Multi-Valued Link field) or a many-to-many relationship (an Association). The Siebel Adapter supports both kinds of relationships.

Siebel applications store the definitions of their business objects and business components in repositories. Different repositories can define different sets of (and properties for) business objects

and business components. A Siebel application can access only the repositories associated with its Object Manager. The repository that defines a business object and business component provides the object's and component's *repository context*.

With the Siebel Adapter, you can configure adapter services that Query, Insert, Update, and Delete business component records. In addition, you can configure services that attach files to business component records, and services that invoke Siebel business services or Siebel business component methods on your Siebel Server. For details, see "webMethods-To-Siebel Communication" on page 71.

For information about Siebel business objects and business components, see the *Tools Guides* on the Siebel Bookshelf.

Siebel Navigation Paths

From a given business component, it is possible to "walk along" the relationships defined for that component to another component. The path you use to traverse component relationships is called the *navigation path*. For example, you might want to obtain all addresses for a particular account. In this case, you can traverse the parent/child relationship between Account and Address to obtain those addresses. By using navigation paths, it is possible to traverse nearly all of the business component relationships defined in the Siebel system.

You can navigate from a top-level business component to any other related component. This defines the navigation path that will be taken to reach the selected component. All operations performed by the adapter will traverse this path prior to performing the selected operation. For example, if you want to select account addresses, select Account as the business object in the tree view. From there, navigate to the Address sub-component by expanding the Account view and selecting the Business Address Multi-Valued Link. By choosing this navigation path, you perform an operation on the Address component for a particular Account.

Because a business component represents all records in a particular database table, it is necessary to identify the particular record or records on which you want to perform an operation when moving down the navigation path. You accomplish this by specifying search criteria at each level of the navigation path. For example, you might want to select the addresses for the Account named Acct1. With the Siebel Adapter, you accomplish this by defining search expressions when you configure the adapter services for your integration.

Occasionally, to properly select the business component instances that will be affected by a given operation, you might need to specify constraints on business components that are not defined along the navigation path (for example, components that exist under the same business object as the selected component, but for which no explicit Multi-Valued Link field relationships exist). You can accomplish this by selecting additional business components under a selected business object. You then can specify search criteria constraints on these components. However, you should almost never need to specify additional business components to add the proper search criteria constraints to your operation. This option is provided for added flexibility. Typically, the use of the navigation path alone should be sufficient.

webMethods-to-Siebel Communication

To use the Siebel Adapter to perform webMethods-to-Siebel communication, you configure adapter services. Adapter services allow you to connect to the adapter's resource and initiate an operation on the resource from the Integration Server.

You call adapter services from flow or Java services to interact with the Siebel Server. The adapter services perform operations by using Siebel's Business Object Interface (BOI) API. Integration Server then uses adapter connections that you defined earlier to execute the adapter services. See "webMethods-To-Siebel Communication" on page 71, for details. The flow or Java services can call adapter services from any kind of Integration Server client, including a Web browser, Java, C/C++, or Visual Basic client. For example, in a browser client you can create an HTML form that interacts with your Siebel connection using the Siebel Adapter. Integration Server and the Siebel Adapter must be on the same network as your Siebel connection. For instructions to invoke services, see the *webMethods Service Development Help* for your release.

Adapter Service Templates

Adapter services are based on adapter service templates, which are provided with the Siebel Adapter. Each template represents a specific technique for doing work on a resource. For example, you use the Query template to retrieve specified information from a Siebel business component. An adapter service template contains all of the code necessary for interacting with the resource but without the data specifications. You provide these specifications when you configure a new adapter service.

To configure a new service, you use Designer. First, you assign the service an adapter connection. Then, you select the adapter service template and supply the data specifications using Designer. Some familiarity with using Designer is required. For more information, see the *webMethods Service Development Help* for your release.

The Siebel Adapter provides the following adapter service templates:

Service Type	Description	Reference page
Query	Performs a Siebel Query operation. It retrieves business component records based on one or more fields.	"Query Services" on page 21
Insert	Performs a Siebel Insert operation. It inserts a business component record with values specified for one or more fields.	"Insert Services" on page 22
Update	Performs a Siebel Update operation. It updates business component records based on the value of one or more fields. It can update one or more record fields.	"Update Services" on page 23

Service Type	Description	Reference page
Delete	Performs a Siebel Delete operation. It deletes business component records based on the value of one or more fields.	"Delete Services" on page 24
Associate	Performs a Siebel Associate operation. This service establishes relationships among multi-value group (MVG) business component records.	"Associate Services" on page 25
Attachment	Creates an attachment file in a business component record, updates an existing attachment, or retrieves an attachment.	"Attachment Services" on page 26
Business Service	Invokes a Siebel business service on your Siebel Server.	"Business Service Services" on page 28
Invoke Business Component Method	Invokes a Siebel business component method on your Siebel Server.	"Invoke Business Component Method Services" on page 29

Multi-Valued Links in Services

When creating a Query, Insert, Update, or Delete service, you can select a multi-valued link (MVL) to a child record if the business component you are using is contained in any MVL relationships. The child fields are in a multi-value group (MVG) business component. When a service returns an MVG business component field value, it returns only one value for the field. Siebel recognizes this value as the primary value. This primary value is always stored in the parent business component.

For example, suppose you want to query the primary business address of a particular account. To do this, you configure a query that uses a multi-valued link to navigate from the Account business component to its child business component, Business Address.

MVL relationships are part of a chain; each business component determines which links are available from it. Not all business components have MVL relationships. The MVL relationships between business components are defined within Siebel and cannot be changed using the adapter.

Changing the Connection Associated With an Adapter Service at Design Time

Integration Server provides a built-in service that you can use at design time to change the connection associated with an adapter service. This built-in service, named setAdapterServiceNodeConnection, is provided in the WmART package's pub.art.service folder. Using this service, you can 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 Designer by selecting the service in one of those tools and running it.

For details, see the *webMethods Integration Server Built-In Services Reference* for your release.

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

Integration Server enables you to dynamically select the connection a service uses to interact with the adapter's resource. This feature enables one service to interact with multiple, similar backend resources.

For example, a service can be defined to use a default connection that interacts with the Siebel Business Object Interface of Siebel Server. However, at run time you can override the default connection and instead use another connection to interact with the Siebel Business Object Interface of another Siebel Server.

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 65.

Passing Credentials at Run Time while Invoking an Adapter Service

With versions of the Siebel Adapter prior to 6.0 SP3, you must configure a connection for each user connecting to the Siebel Server by providing individual user name and password credentials for each connection. For example, if multiple users needed to connect to the same Siebel Server, you were required to configure one connection for each user at design time.

With Siebel Adapter 6.0 SP3, you can dynamically provide the user name and password credentials associated with a specific adapter service at run time. This capability enables you to override the connection that was associated with the adapter service at design time. If you provide the user name and password credentials in an adapter service at run time, the Siebel Adapter connects to the Siebel Server using the new credentials, along with the other connection parameters associated with the service's associated connection. If you do not provide any user credentials at run time, the adapter service is invoked using the user credentials provided at design time.

Note:
To retrieve information from the Siebel Server when configuring adapter services, the Siebel Adapter uses the connection parameters in the service's associated connection.

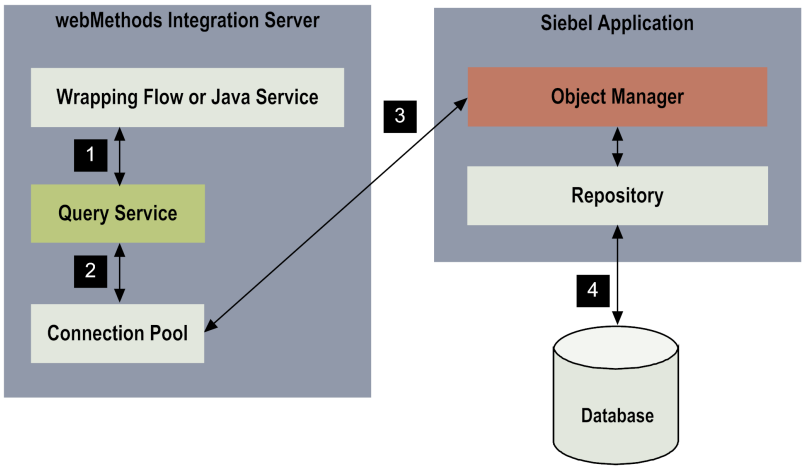
Query Services

A Query service performs a Siebel Query operation. It retrieves business component records based on one or more fields.

To configure Query services, see "Configuring Query, Insert, Update, and Delete Services" on page 73.

Query Service Run-Time Processing

The following diagram illustrates how the Siebel Adapter processes a Query service at run time.



Step	Description
1	A flow or Java service, typically invoked by an Integration Server client, initiates the Query service on the Integration Server. The client passes the required input information to the service. You configure the Query service and the wrapping flow or Java service using Designer.
2	The adapter service retrieves a connection from the service's associated connection pool. You configure and enable the adapter connection using the Integration Server Administrator. Adapter connections contain attributes that help the adapter connect to the Siebel application. For more information about connection pooling, see "Connection Pooling" on page 14.
3	The Object Manager performs the Query service's operation, based on the search criteria specified in the service. The service uses Siebel's Business Object Interface (BOI) API to process the operation.
4	If the operation is successful, the Object Manager passes the result to the Query service, which returns the result to the client.

Step	Description
	If the operation is unsuccessful, the Object Manager returns an error to the adapter, which passes the exception to the Integration Server logs. For more information about how the adapter handles exceptions, see "Logging and Exception Handling" on page 113.

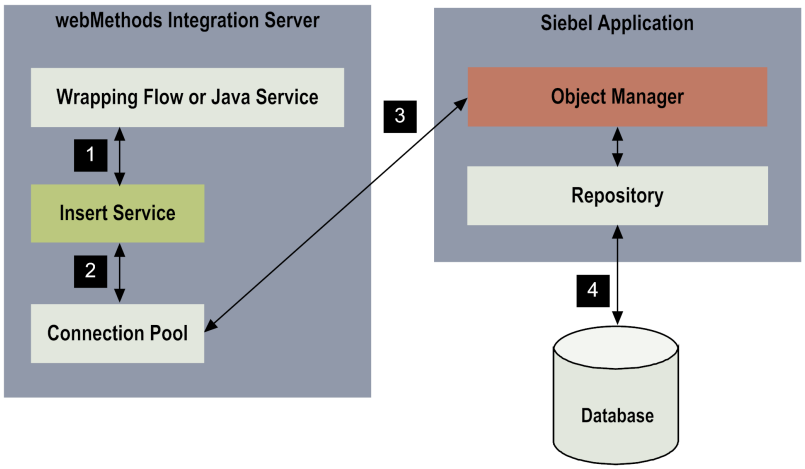
Insert Services

An Insert service performs a Siebel Insert operation. It inserts a business component record with values specified for one or more fields.

To configure Insert services, see "Configuring Query, Insert, Update, and Delete Services" on page 73.

Insert Service Run-Time Processing

The following diagram illustrates how the Siebel Adapter processes an Insert service at run time.



Step	Description
1	A flow or Java service, typically invoked by an Integration Server client, initiates the Insert service on the Integration Server. The client passes the required input information to the service. You configure the Insert service and the wrapping flow or Java service using Designer.
2	The adapter service retrieves a connection from the service's associated connection pool. You configure and enable the adapter connection using the Integration Server Administrator. Adapter connections contain attributes that help the adapter connect to the Siebel application. For more information about connection pooling, see "Connection Pooling" on page 14.

Step	Description
3	The Object Manager performs the Insert service's operation, based on the search criteria specified in the service. The service uses Siebel's Business Object Interface (BOI) API to process the operation.
4	If the operation is successful, the Object Manager passes the result to the Insert service, which returns the result to the client. The result is a string field that indicates the row ID of the row inserted by the service. If the operation is unsuccessful, the Object Manager returns an error to the adapter, which passes the exception to the Integration Server logs. For more information about how the adapter handles exceptions, see "Logging and Exception Handling" on page 113.

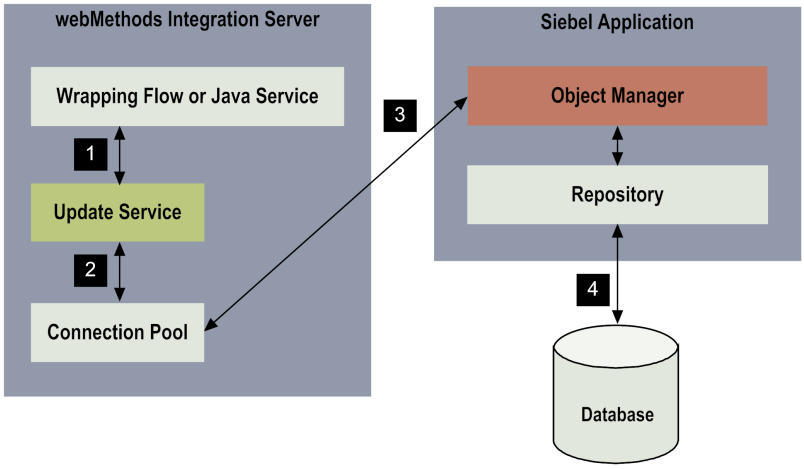
Update Services

An Update service performs a Siebel Update operation. It updates business component records based on the value of one or more fields. It can update one or more record fields.

To configure Update services, see "Configuring Query, Insert, Update, and Delete Services" on page 73.

Update Service Run-Time Processing

The following diagram illustrates how the Siebel Adapter processes an Update service at run time.



Step	Description
1	A flow or Java service, typically invoked by an Integration Server client, initiates the Update service on the Integration Server. The client passes the required input information to the service. You configure the Update service and the wrapping flow or Java service using Designer.

Step	Description
2	The adapter service retrieves a connection from the service's associated connection pool. You configure and enable the adapter connection using the Integration Server Administrator. Adapter connections contain attributes that help the adapter connect to the Siebel application. For more information about connection pooling, see "Connection Pooling" on page 14.
3	The Object Manager performs the Update service's operation, based on search criteria specified in the service. The service uses Siebel's Business Object Interface (BOI) API to process the operation.
4	If the operation is successful, the Object Manager passes the result to the Update service, which returns the result to the client. The result is a string field that indicates the number of rows updated by the service. If the operation is unsuccessful, the Object Manager returns an error to the adapter, which passes the exception to the Integration Server logs. For more information about how the adapter handles exceptions, see "Logging and Exception Handling" on page 113.

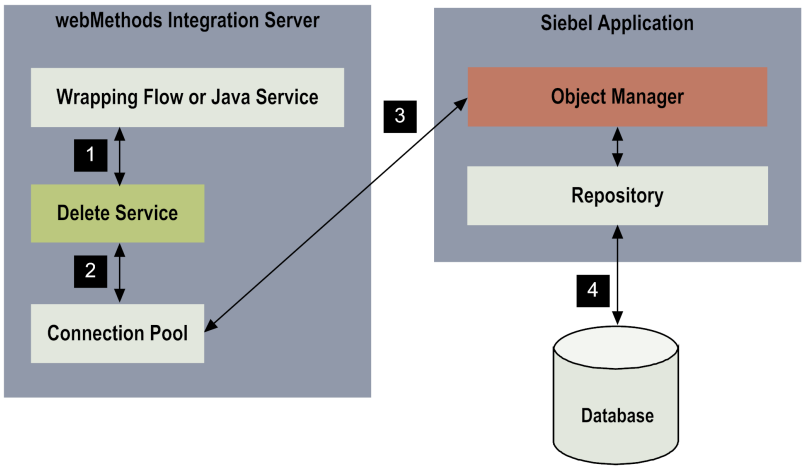
Delete Services

A Delete service performs a Siebel Delete operation. It deletes business component records based on the value of one or more fields.

To configure Delete services, see "Configuring Query, Insert, Update, and Delete Services" on page 73.

Delete Service Run-Time Processing

The following diagram illustrates how the Siebel Adapter processes a Delete service at run time.



Step	Description
1	A flow or Java service, typically invoked by an Integration Server client, initiates the Delete service on the Integration Server. The client passes the required input information to the service. You configure the Delete service and the wrapping flow or Java service using Designer.
2	The adapter service retrieves a connection from the service's associated connection pool. You configure and enable the adapter connection using the Integration Server Administrator. Adapter connections contain attributes that help the adapter connect to the Siebel application. For more information about connection pooling, see "Connection Pooling" on page 14.
3	The Object Manager performs the Delete service's operation, based on search criteria specified in the service. The service uses Siebel's Business Object Interface (BOI) API to process the operation.
4	If the operation is successful, the Object Manager passes the result to the Delete service, which returns the result to the client. The result is a string field that indicates the number of rows deleted by the service. If the operation is unsuccessful, the Object Manager returns an error to the adapter, which passes the exception to the Integration Server logs. For more information about how the adapter handles exceptions, see "Logging and Exception Handling" on page 113.

Associate Services

An Associate service performs a Siebel Associate operation. This service establishes relationships among multi-value group (MVG) business component records.

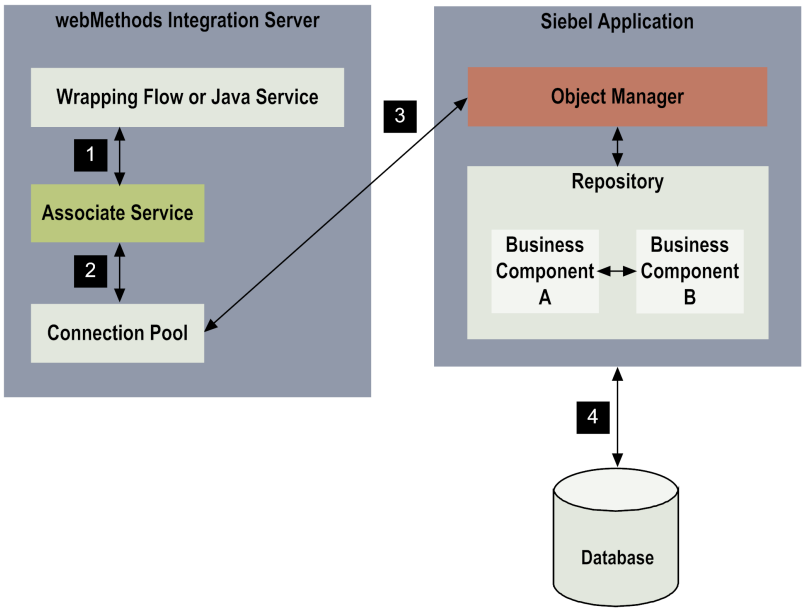
These associations have a many-to-many relationship. For example, if you are creating orders, and you want to associate line items with the multi-valued link field in the orders, you would associate the line items with the order.

Each level in the navigation path must return a single record. You can associate only those component records that are linked by an association relationship in Siebel.

To configure Associate services, see "Configuring Associate Services" on page 78.

Associate Service Run-Time Processing

The following diagram illustrates how the Siebel Adapter processes an Associate service at run time.



Step	Description
1	A flow or Java service, typically invoked by an Integration Server client, initiates the Associate service on the Integration Server. The client passes the required input information to the service. You configure the Associate service and the wrapping flow or Java service using Designer.
2	The adapter service retrieves a connection from the service's associated connection pool. You configure and enable the adapter connection using the Integration Server Administrator. Adapter connections contain attributes that help the adapter connect to the Siebel application. For more information about connection pooling, see "Connection Pooling" on page 14.
3	The Object Manager performs the Associate service's operation. That is, it associates the business components specified in the service. The service uses Siebel's Business Object Interface (BOI) API to process the operation.
4	If the operation is successful, the Object Manager passes the result to the Associate service, which returns the result to the client. The output is a string field that indicates the number of rows of the associated MVG business component record. If the operation is unsuccessful, the Object Manager returns an error to the adapter, which passes the exception to the Integration Server logs. For more information about how the adapter handles exceptions, see "Logging and Exception Handling" on page 113.

Attachment Services

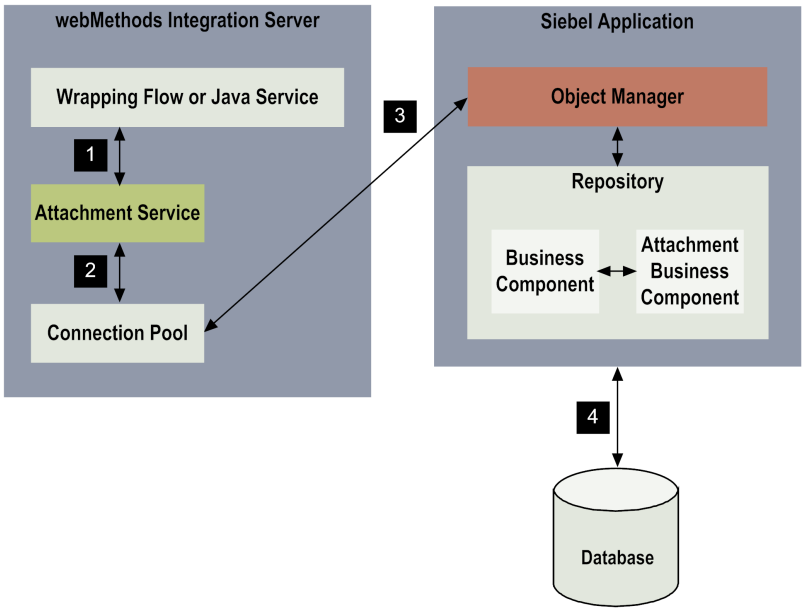
An Attachment service creates an attachment file in a business component record, updates an existing attachment, or obtains an attachment. To do this, at run time you specify that the

Attachment service invokes one of the following Siebel methods: CreateFile, PutFile, or GetFile, respectively.

To configure Attachment services, see "Configuring Attachment Services" on page 81.

Attachment Service Run-Time Processing

The following diagram illustrates how the Siebel Adapter processes an Attachment service at run time.



Step	Description
1	A flow or Java service, typically invoked by an Integration Server client, initiates the Attachment service on the Integration Server. The client passes the required input information to the service. You configure the Attachment service and the wrapping flow or Java service using Designer.
2	The adapter service retrieves a connection from the service's associated connection pool. You configure and enable the adapter connection using the Integration Server Administrator. Adapter connections contain attributes that help the adapter connect to the Siebel application. For more information about connection pooling, see "Connection Pooling" on page 14.
3	The Object Manager performs the Attachment service's operation. That is, it creates, updates, or obtains the attachment file specified in the service, using Siebel's Business Object Interface (BOI) API to process the operation.
4	If the operation is successful, the Object Manager passes the result to the Attachment service, which returns the result to the client. The output is a string field that indicates

Step	Description
	Success or Failure. If the service invoked a GetFile method, the field also indicates the path of the attachment returned by the service. If the operation is unsuccessful, the Object Manager returns an error to the adapter, which passes the exception to the Integration Server logs. For more information about how the adapter handles exceptions, see "Logging and Exception Handling" on page 113.

Business Service Services

A Business Service service invokes a Siebel business service on your Siebel Server.

Siebel business services are reusable code modules that perform specified functions while running in the Siebel Object Manager. For example, a business service might perform a tax calculation function. Siebel provides several predefined business services. Alternatively, you can write your own business services using Siebel VB or Siebel eScript.

You can trigger or invoke business services from various contexts within the Siebel environment, including user interface events, Siebel Workflow actions, inbound message requests, and from external applications.

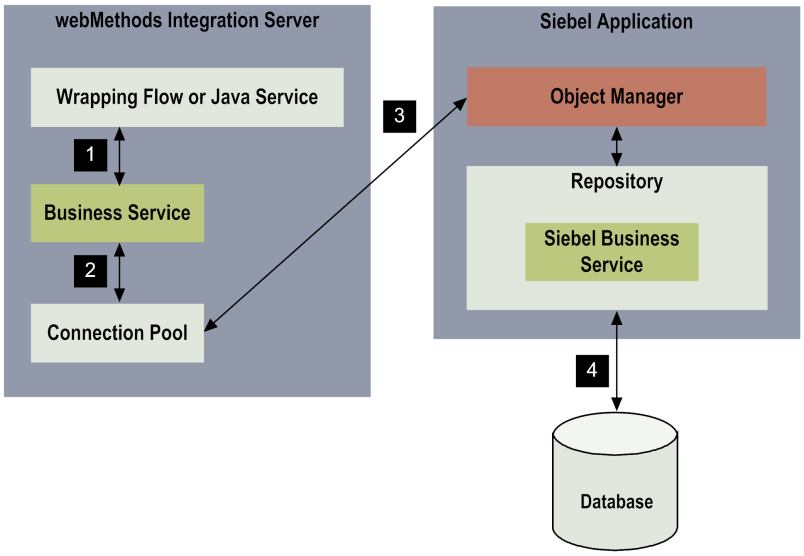
Note the following:

- The data that the Integration Server passes to the Siebel business service must be of the data type String. This data must not be nested. If you need to pass nested data, add the webMethods docToString flow service to your adapter service to convert the nested data into a single string before passing the data. To recreate the nested data in Siebel, add the appropriate logic to your Siebel business service.
- The output that the Siebel business service returns cannot contain nested data. If you need to pass nested data to the Integration Server, use appropriate logic in the Siebel business service to convert the data into a single string before returning the data to the Integration Server. When the adapter service receives the data, use the webMethods stringToDoc flow service to recreate the original nested data structure.

To configure Business Service services, see "Configuring Services That Invoke Siebel Business Services" on page 86.

Business Service Service Run-Time Processing

The following diagram illustrates how the Siebel Adapter processes a Business Service service at run time.



Step	Description
1	A flow or Java service, typically invoked by an Integration Server client, initiates the Business Service service on the Integration Server. The client passes the required input information to the service. You configure the Business Service service and the wrapping flow or Java service using Designer.
2	The adapter service retrieves a connection from the service's associated connection pool. You configure and enable the adapter connection using the Integration Server Administrator. Adapter connections contain attributes that help the adapter connect to the Siebel application. For more information about connection pooling, see "Connection Pooling" on page 14.
3	The Object Manager invokes the business service specified in the Business Service service, using Siebel's Business Object Interface (BOI) API to process the invocation.
4	If the operation is successful, the Object Manager passes the result of the Siebel business service to the Business Service service, which returns the result to the client. If the operation is unsuccessful, the Object Manager returns an error to the adapter, which passes the exception to the Integration Server logs. For more information about how the adapter handles exceptions, see "Logging and Exception Handling" on page 113.

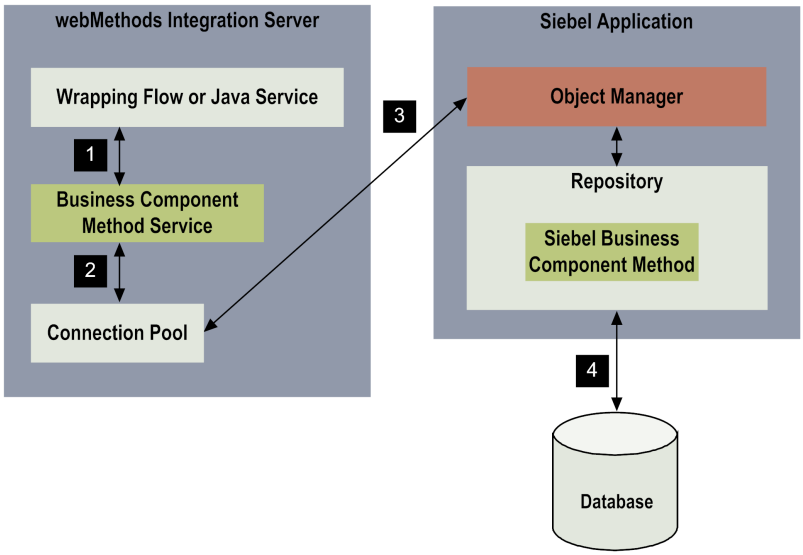
Invoke Business Component Method Services

An Invoke Business Component Method service invokes a Siebel business component method on your Siebel Server.

To configure Invoke Business Component Method services, see "Configuring Services That Invoke Business Component Methods" on page 89.

Invoke Business Component Method Service Run-Time Processing

The following diagram illustrates how the Siebel Adapter processes an Invoke Business Component Method service at run time.



Step	Description
1	A flow or Java service, typically invoked by an Integration Server client, initiates the Business Service service on the Integration Server. The client passes the required input information to the service. You configure the Business Service service and the wrapping flow or Java service using Designer.
2	The adapter service retrieves a connection from the service's associated connection pool. You configure and enable the adapter connection using the IIntegration Server Administrator. Adapter connections contain attributes that help the adapter connect to the Siebel application. For more information about connection pooling, see "Connection Pooling" on page 14.
3	The Object Manager invokes the business component method specified in the Invoke Business Component Method service, using Siebel's Business Object Interface (BOI) API to process the invocation.
4	If the operation is successful, the Object Manager passes the result to the Invoke Business Component Method service, which returns the result to the client. The output is a string field named Status, which contains the output generated by the method that is invoked. If the operation is unsuccessful, the Object Manager returns an error to the adapter, which passes the exception to the Integration Server logs. For more information about how the adapter handles exceptions, see "Logging and Exception Handling" on page 113.

Siebel-to-webMethods Communication

There are three ways to accomplish Siebel-to-webMethods communication:

- EAI webMethods Transports
- Siebel Visual Basic (VB) scripts
- HTTP posting to the Integration Server using the Siebel business service named EAI HTTP Transport

The EAI webMethods Transports offer more functionality and greater efficiency than either VB scripts or HTTP posting. The following table compares the features of all three.

Feature	Transports	VB Scripts	EAI HTTP Transport
Supported on Windows?	X	X	X
Supported on HP-UX and Solaris?	X		X
Usable in Siebel business component scripts?	X	X	
Usable in Siebel workflows?	X		X
Supports guaranteed delivery of data?	X		
Supports both synchronous and asynchronous service invocation?	X		
Supports simultaneous use by multiple users?	X		X
Reduces network load by passing data in non-XML format?	X		
Scalable?	X		

For more information, see these sections:

- "EAI webMethods Transports" on page 31
- "Siebel Visual Basic Scripts" on page 32
- "EAI HTTP Transport" on page 34

EAI webMethods Transports

The EAI webMethods Transports are custom Siebel business service methods. They enable Siebel to notify Integration Server services when a change (such as an Insert, Update, or Delete) occurs within a particular Siebel business component. Transports are designed to provide interactive transportation of small volumes of data between the Siebel application and the Integration Server.

Transports are built using Siebel's eScript scripting language. To see an example function that invokes a transport, see "Example Transport Invocation Function" on page 109.

To test transports, you can use the Siebel Business Service Simulator. For an example, see "Testing Synchronous IS Service Invoke" on page 100.

Transports run on the Siebel Server, using Java Data Beans to connect to the Siebel Server. You can call transports from within a Siebel business component script or from a Siebel workflow. Calling transports from workflows allows you to easily update parameters for the transports.

You can use transports to perform many tasks, including synchronously or asynchronously invoking a service on the Integration Server, getting a transaction's status or data, restarting or ending a transaction, and reporting exceptions. For more information, see "EAI webMethods Transports" on page 98.

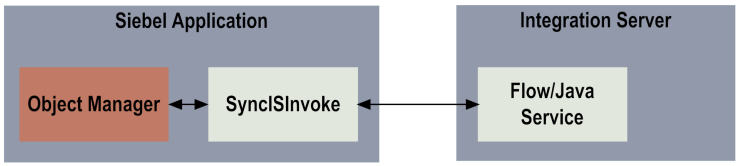
Note:
EAI webMethods Transports are supported on Siebel 7.x on Windows, HP-UX, and Solaris platforms. Before you can use transports, you must configure your system as described in "Installing, Upgrading, and Uninstalling the Siebel Adapter " on page 37.

Note the following:

- The data returned by an Integration Server service must be of the data type String.
- The data returned by an Integration Server service must not be nested. If you need to return nested data, add to your Integration Server service the webMethods flow service docToString to convert the incoming data into a single string before returning the data. The transport's outputData child property set will contain this string when the Invoke operation is completed. You can retrieve the data (which is in the form of a string) from outputData.

EAI webMethods Transport Run-Time Processing

The following diagram shows how the transport method Synchronous IS Service Invoke in the Siebel application invokes a flow or Java service on the Integration Server. The service may return data to the transport, using the Object Manager to complete the processing.



For more information about using transports, see "EAI webMethods Transports" on page 98.

Siebel Visual Basic Scripts

The Siebel Event Model enables you to attach procedures to Siebel objects, such as business components and applets. You can specify that when a particular event is triggered, such as when users modify a business component record or click an applet control, the attached procedure runs.

You write procedures using Siebel's scripting language, Siebel Visual Basic (VB). Because Siebel VB is compatible with Microsoft's Visual Basic, you can write procedures that can interact with COM objects. The COM interface, enables Siebel VB to call adapter services and pass values to them. For example, you might create a Siebel procedure that notifies an Integration Server client when a new sales contact is created.

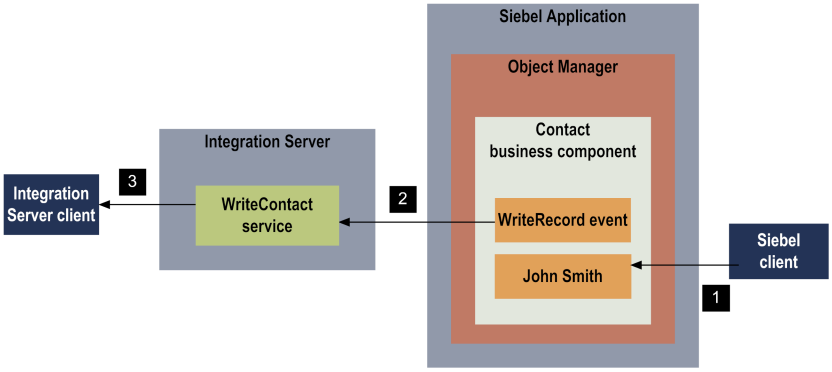
If you want a GUI event to call adapter services, you can attach a procedure to a control in a Siebel application GUI applet. For example, you could attach a Siebel VB procedure to a Siebel application applet's **OK** button. When users click the **OK** button, the Siebel VB procedure then invokes an adapter service.

If you want some kind of business component processing to trigger an event that calls adapter services, you might place the script within one of the predefined procedures that are attached to business components. For example, you might place a script in the Contact business component's WriteRecord event.

For information about writing a Siebel VB script, see "Siebel Visual Basic Scripts" on page 110.

Siebel VB Script Run-Time Processing

The following diagram shows the run-time processing of the example mentioned above.



Step	Description
1	A Siebel client adds a new Contact business component record named John Smith.
2	The Siebel event associated with the Contact business component, WriteRecord, is triggered. WriteRecord's Siebel VB procedure calls the adapter service WriteContact, and passes input values as a Values object.
3	The adapter service WriteContact processes and sends a notification to the Integration Server client notifying it of the addition of John Smith.

For more information about writing Siebel VB scripts, see "Siebel Visual Basic Scripts" on page 110.

EAI HTTP Transport

The EAI HTTP Transport business service enables your Siebel application to post XML documents or flat files to the Integration Server. Using this posting service is a viable solution, but it is not as scalable and secure as the webMethods transports because the service:

- Does not provide the Quality of Service (QoS) that the webMethods transports provide, such as security, guaranteed delivery, failover support, clustering, connection management, and thread management.
- Uses an additional layer of transformation between an XML document and an Integration Server record.
- Uses an additional layer of transformation between an XML document, a Siebel integration object, a Siebel business object, and a Siebel business component.

For information about using the EAI HTTP Transport business service, see your Siebel documentation. For information about how the Integration Server can retrieve data for services, see the *webMethods Integration Server Administrator's Guide* for your release.

Using Version Control Systems to Manage Adapter Elements

The adapter supports the Version Control System (VCS) Integration feature provided by Designer. When you enable the feature in Integration Server, you can check adapter packages or elements into and out of your version control system from Designer. For more information about the VCS Integration feature, see the *Configuring the VCS Integration Feature*.

Beginning with Integration Server 8.2 SP3, the adapter supports the local service development feature in Designer. This feature extends the functionality of the VCS Integration feature to check package elements and their supporting files into and out of a VCS directly from Designer. For more information about local service development and how it compares to the VCS Integration feature, see the *webMethods Service Development Help*.

Optimize Infrastructure Data Collector Support for the Adapter

Optimize 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 other external tools. When you start monitoring an Integration Server, Optimize 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 *Administering webMethods Optimize* for your release.

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 *webMethods Integration Server Administrator's Guide* for your release.

2 Installing, Upgrading, and Uninstalling the Siebel Adapter

Overview

This chapter explains how to install, upgrade, and uninstall webMethods Siebel Adapter 6.0 SP3. The instructions use the Software AG Installer and the Software AG Uninstaller wizards. For complete information about the wizards or other installation methods, or to install other webMethods products, see the *Installing webMethods Products On Premises* for your release.

Requirements

For a list of the operating systems and Siebel products supported by Siebel Adapter 6.0 SP3, see the *webMethods Adapters System Requirements* .

Siebel Adapter 6.0 SP3 requires the Siebel components listed below; see the *webMethods Adapters System Requirements* for supported versions.

Component	Comment
Siebel Server	The Siebel Server can be on a different machine than the Siebel Adapter.
Siebel Client	A supported Siebel client must be installed on the same machine as the Siebel Adapter. If you use a Siebel Java Data Bean client, obtain the following .jar files from the Siebel Server installation CD and place them in the wmSiebelAdapter\code\jars directory: ■ Pre-7.1.1 Siebel Clients - SiebelJI_Common.jar ■ Siebel Client 7.7.1 or later - Siebel.jar ■ SiebelJI_*locale*.jar (for example, SiebelJI_enu.jar for English)

Siebel Adapter 6.0 SP3 has no hardware requirements beyond those of the host Integration Server.

Installing Siebel Adapter 6.0 SP3

Note:
If you are installing the Siebel Adapter in a clustered environment, you must install the adapter 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 "Package Management" on page 51.

To install the webMethods Siebel Adapter

1. Siebel Adapter 6.0 SP3 provides a new feature that lets you dynamically provide the user name and password credentials associated with a specific adapter service at run time. To use this feature, apply the appropriate fixes below:

Integration Server	Fixes
6.5	ID_6-5_Fix2 and IS_ART_6-5_Fix7
6.1	ID_6-1_SP1_Fix7 and IS_ART_6-1_SP1_Fix7
6.0	ID_6-0-1_SP1_Fix11 and IS_ART_6-0-1_SP4_Fix4

For instructions on using this feature, see "Overview of the Adapter" on page 9.

2. If you are going to use Siebel Adapter 6.0 SP3 with Siebel Server 8.0, also apply fix SBL_6-0_SP3_Fix8 for Siebel Server 8.0.

3. Download Software AG Installer from the Empower Product Support Web site.

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

5. Start Software AG Installer.

   - Choose the webMethods release that includes the Integration Server on which you want to install the adapter. For example, if you want to install the adapter on Integration Server 7.1, choose the 7.1 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.
   - In the product selection list, select **Adapters > webMethods Siebel Adapter 6.0 > Service Pack 3 > Program Files**. You can also choose to install documentation and the Siebel Server Component, which contains the EAI webMethods Transports required for Siebel-to-webMethods connections.

6. After installation is complete, start the host Integration Server.

   When starting the adapter, you might receive a local encoding-related error message similar to the following:

   ```
   Error: ... threw a SiebelException: code(8210), msg(Code Page "8859_1" is not
   supported. Use "-encoding" option to change the file encoding.)
   ```

   To fix this problem, append an appropriate value to the -Dfile.encoding parameter in the JAVA_MEMSET variable in your server.bat or server.sh file. For example:

   ```
   set JAVA_MEMSET=-ms%JAVA_MIN_MEM% -mx%JAVA_MAX_MEM% -Dfile.
   encoding="ISO8859-1"
   ```

   For a complete list of encoding values, see your Java documentation.

7. Configure your system to use the Transports, as described in "Configuring Your System to Use the EAI webMethods Transports" on page 40.

webMethods Transport Files in Siebel Adapter 6.0 SP3

With Siebel Adapter 6.0 SP3, the Siebel transport sif file for pre-7.7.1 Siebel Servers is also available. For more information on the Siebel Server versions supported by the Siebel Adapter, see "Requirements" on page 38.

The Siebel transport sif files are different, depending on the Siebel Server versions:

- For Siebel Server version 7.7.1 and later, the sif file is WmISTransport.sif.
- For Siebel Server version prior to 7.7.1, the sif file is WmISTransport_Pre7.7.1.sif.

Both the files are located in the bin\wmTransport folder in the *Integration Server_directory*\WmSiebelAdapter\backEnd\WmISTransport<*platform*>.zip file.

Configuring Your System to Use the EAI webMethods Transports

For Siebel-to-webMethods communications to function, you must install the EAI webMethods Transports (Siebel Server Component). You must then configure your system to use the transports using the appropriate procedure below.

Note:
If you are using Siebel Server 7.7 or later, you must add Siebel.jar to the CLASSPATH. You do not need to add the SiebelJI_Common.jar file.

Windows

To configure your Windows system

1. Stop your Siebel Server.
2. Copy the following files to the `TransportsRoot_directory`\bin\wmTransport directory:

For Integration Server...	Copy these jar files...
7.1 and later	`installation_directory\common\lib\wm-isclient.jar`
	`installation_directory\IntegrationServer\lib\wm-isserver.jar`
	`installation_directory\common\lib\ext\enttoolkit.jar`
	`installation_directory\common\lib\ext\mail.jar`
7.0 and earlier	`installation_directory\lib\client.jar`
	`installation_directory\IntegrationServer\lib\server.jar`

For Integration Server...	Copy these jar files...
	`installation_directory\common\lib\ext\enttoolkit.jar`
	`installation_directory\common\lib\ext\mail.jar`

TransportsRoot_directory and *installation_directory* are the directories in which you installed the EAI webMethods Transports and the webMethods products, respectively.

3. Open the `TransportsRoot_directory\bin\wmTransport\wmISTransportCfg.txt` file and assign values to its variables as indicated in the file's comments.

4. Modify your system variables using the Windows Control Panel as follows:

   a. Modify the PATH variable as follows:

For...	Modify the PATH as follows...

JDK 1.4 and later	`PATH=jdk_install_dir\bin;jdk_install_dir\jre\bin;jdk_install_dir\jre\bin\client;%PATH%`
	
	where *jdk_install_dir* is the directory in which you installed the Java Development Kit. For example, if you installed JDK in the c:\jdk1.5 directory, your PATH statement would be:
	
	`PATH=C:\jdk1.5\bin;C:\jdk1.5\jre\bin;C:\jdk1.5\jre\bin\client;%PATH%`

JDK 1.3 and earlier	`PATH=jdk_install_dir\bin;jdk_install_dir\jre\bin;jdk_install_dir\jre\bin\classic;%PATH%`
	
	where *jdk_install_dir* is the directory in which you installed the Java Development Kit. For example, if you installed JDK in the c:\jdk1.3.1 directory, your PATH statement would be:
	
	`PATH=C:\jdk1.3.1\bin;C:\jdk1.3.1\jre\bin;C:\jdk1.3.1\jre\bin\classic;%PATH%`

Note:
With Integration Server 7.1 and later, wmISTransport can be run only on JDK 1.5 and later.

   b. Add the WEBMTCFG variable as follows:

```
WEBMTCFG=TransportsRoot_directory\bin\wmTransport
```

   c. Modify the CLASSPATH variable to add the jar files as follows:

For Integration Server...	The jar files are...
7.1 and later	`CLASSPATH=%CLASSPATH%;TransportsRoot_directory\bin\wmTransport\wmISTransport.jar;TransportsRoot_directory\bin\wmTransport\enttoolkit.jar; TransportsRoot_directory\bin\wmTransport\wm-isclient.jar; TransportsRoot_directory\bin\wmTransport\wm-isserver.jar; TransportsRoot_directory\bin&\wmTransport\mail.jar`
7.0 and earlier	`CLASSPATH=%CLASSPATH%;TransportsRoot_directory\bin\wmTransport\wmISTransport.jar; TransportsRoot_directory\bin\wmTransport\enttoolkit.jar; TransportsRoot_directory\bin\wmTransport\client.jar; TransportsRoot_directory\bin\wmTransport\server.jar; TransportsRoot_directory\bin\wmTransport\mail.jar`

5. Import `TransportsRoot_directory`\bin\wmTransport\wmISTransport.sif into your Siebel Server's Repository using Siebel Tools.

   Note:
   If you are using a version of Siebel Server prior to 7.7.1, the sif file name is WmISTransport_Pre7.7.1.sif. For more information, see " webMethods Transport Files in Siebel Adapter 6.0 SP3" on page 40.

6. Compile your Siebel repository. Make sure you have Write access to your Siebel Server's siebel.srf file.

7. Restart your Siebel Server.

HP-UX

To configure your HP-UX system

1. Stop your Siebel Server.

2. Copy the necessary jar files from *installation_directory*/common/lib to *TransportsRoot_directory*/bin directory:

   - For Integration Server 7.1 and later, the jar file is wm-isclient.jar.
   - For Integration Server 7.0 and earlier, the jar file is client.jar.

3. Open the *TransportsRoot_directory*/bin/wmTransport/wmISTransportCfg.txt file and assign values to its variables as indicated in the file's comments.

4. Open the *SiebelServer_directory*/siebsrvr/siebenv.sh file and modify or add the following variables at the end of the file as follows:

a. Modify the CLASSPATH statement as shown below:

For Integration Server...	The CLASSPATH statement looks like...
7.1 and later	`CLASSPATH=${CLASSPATH}:${SIEBEL_ROOT}/bin/wmTransport/wmISTransport.jar:${SIEBEL_ROOT}/bin/wmTransport/wmisclie.nt.jar:${SIEBEL_ROOT}/bin/wmTransport/SiebelJI_Common.jar:${SIEBEL_ROOT}/bin/wmTransport/Siebel.jar:${SIEBEL_ROOT}/bin/wmTransport/SiebelJI_enu.jar ; export CLASSPATH`
7.0 and earlier	`CLASSPATH=${CLASSPATH}:${SIEBEL_ROOT}/bin/wmTransport/wmISTransport.jar:${SIEBEL_ROOT}/bin/wmTransport/client.jar:${SIEBEL_ROOT}/bin/wmTransport/SiebelJI_Common.jar:${SIEBEL_ROOT}/bin/wmTransport/Siebel.jar:${SIEBEL_ROOT}/bin/wmTransport/SiebelJI_enu.jar ; export CLASSPATH`

b. Add the transport:

```
WEBMTCFG=${SIEBEL_ROOT}/bin/wmTransport ; export WEBMTCFG
```

c. Modify the PATH statement:

For Integration Server...	Use...
7.1 and later	JRE 1.5 or higher: `PATH=${PATH}:/opt/java1.5/bin:/opt/java1.5/jre/bin:/opt/java1.5/jre/ bin/PA_RISC ; export PATH`
7.0 and earlier	The JVM that is installed with Integration Server. For example: `PATH=${PATH}:/opt/java1.3/bin:/opt/java1.3/jre/bin:/opt/java1.3/jre/ bin/PA_RISC ; export PATH`

d. Modify the library paths and heap size statements:

For Integration Server...	Use...
7.1 and later	JRE 1.5 or higher: `SHLIB_PATH=${SHLIBPATH}:${SIEBEL_ROOT}/bin:${SIEBEL_ROOT}/lib:/lib:/usr/lib:/opt/java1.5/jre/lib/PA_RISC:/opt/java1.5/jre/lib/PA_RISC/classic ; export` `SHLIB_PATH LD_LIBRARY_PATH=${SHLIB_PATH} ; export`

For Integration Server...	Use...
	`LD_LIBRARY_PATH IST_MIN_HEAP_SIZE=128M ; export IST_MIN_HEAP_SIZE IST_MAX_HEAP_SIZE=128M ; export IST_MAX_HEAP_SIZE`
7.0 and earlier	The JVM that is installed with Integration Server. For example: `SHLIB_PATH=${SHLIBPATH}:${SIEBEL_ROOT}/bin:${SIEBEL_ROOT}/lib:/lib:/usr/lib:/opt/java1.3/jre/lib/PA_RISC:/opt/java1.3/jre/lib/PA_RISC/ classic ; export SHLIB_PATH LD_LIBRARY_PATH=${SHLIB_PATH} ; export LD_LIBRARY_PATH IST_MIN_HEAP_SIZE=128M ; export IST_MIN_HEAP_SIZE IST_MAX_HEAP_SIZE=128M ; export IST_MAX_HEAP_SIZE`

e. Save and close the file.

5. Import *TransportsRoot_directory*/bin/wmTransport/wmISTransport.sif into your Siebel Server's repository using Siebel Tools.

Note:
If you are using a version of Siebel Server prior to 7.7.1, the sif file name is WmISTransport_Pre7.7.1.sif. For more information, see " webMethods Transport Files in Siebel Adapter 6.0 SP3" on page 40.

6. Compile your Siebel repository. Make sure you have Write access to your Siebel Server's siebel.srf file.

7. Execute the siebenv.sh file that you edited earlier.

8. Restart your Siebel Server.

Solaris

To configure your Solaris system

1. Stop your Siebel Server.

2. Copy the necessary jar files from *installation_directory*/common/lib to *TransportsRoot_directory*/bin directory:

   - For Integration Server 7.1 and later, the jar file is wm-isclient.jar.
   - For Integration Server 7.0 and earlier, the jar file is client.jar.

3. Open the *TransportsRoot_directory*/bin/wmTransport/wmISTransportCfg.txt file and assign values to its variables as indicated in the file.

4. Open the *SiebelServer_directory*/siebsrvr/siebenv.sh file and modify or add the following variables at the end of the file as follows:

   a. Modify the CLASSPATH statement as shown below:

For Integration Server...	The CLASSPATH statement looks like...
7.1 and later	`CLASSPATH=${CLASSPATH}:${SIEBEL_ROOT}/bin/wmTransport /wmISTransport.jar:${SIEBEL_ROOT}/bin/ wmTransport/wmisclient.jar:${SIEBEL_ROOT}/bin/ wmTransport/SiebelJI_Common.jar:${SIEBEL_ROOT}/bin/ wmTransport/Siebel.jar:${SIEBEL_ROOT}/bin/wmTransport /SiebelJI_enu.jar ; export CLASSPATH`
7.0 and earlier	`CLASSPATH=${CLASSPATH}:${SIEBEL_ROOT}/bin/wmTransport/ wmISTransport.jar:${SIEBEL_ROOT}/bin/ wmTransport/client.jar:${SIEBEL_ROOT}/bin/ wmTransport/SiebelJI_Common.jar:${SIEBEL_ROOT}/bin/ wmTransport/Siebel.jar:${SIEBEL_ROOT}/bin/wmTransport/ SiebelJI_enu.jar ; export CLASSPATH`

   b. Add the transport:

```
WEBMTCFG=${SIEBEL_ROOT}/bin/wmTransport ; export WEBMTCFG
```

   c. Modify the PATH statement:

For Integration Server...	Use...
7.1 and later	JRE 1.5 or higher: `PATH=${PATH}:/opt/java1.5/bin:/opt/java1.5/jre/bin:/ opt/java1.5/jre/bin/PA_RISC ; export PATH`
7.0 and earlier	the JVM that is installed with Integration Server. For example: `PATH=${PATH}:/opt/javal.3/bin:/opt/java1.3/jre/bin/ PA_RISC ; export PATH`

   d. Modify the library and heap size statements.

For Integration Server...	Use...
7.1 and later	JRE 1.5 or higher:

For Integration Server...	Use...
	`SHLIB_PATH=${SHLIBPATH}:${SIEBEL_ROOT}/bin:${SIEBEL_ROOT}/lib:/lib:/usr/lib:/opt/java1.3/jre/lib/PA_RISC:/opt/java1.3/jre/lib/PA_RISC/classic ; export SHLIB_PATH LD_LIBRARY_PATH=${SHLIB_PATH} ; export LD_LIBRARY_PATH IST_MIN_HEAP_SIZE=128M ; export IST_MIN_HEAP_SIZE IST_MAX_HEAP_SIZE=128M ; export IST_MAX_HEAP_SIZE`
7.0 and later	the JVM that is installed with Integration Server. For example: `SHLIB_PATH=${SHLIBPATH}:${SIEBEL_ROOT}/bin:${SIEBEL_ROOT}/lib:/lib:/usr/lib:/opt/java1.3/jre/lib/PA_RISC:/opt/java1.3/jre/lib/PA_RISC/ classic ; export SHLIB_PATH LD_LIBRARY_PATH=${SHLIB_PATH} ; export LD_LIBRARY_PATH IST_MIN_HEAP_SIZE=128M ; export IST_MIN_HEAP_SIZE IST_MAX_HEAP_SIZE=128M ; export IST_MAX_HEAP_SIZE`

e. Save and close the file.

5. Import *TransportsRoot_directory*/bin/wmTransport/wmISTransport.sif into your Siebel Server's repository using Siebel Tools.

   Note:
   If you are using a version of Siebel Server prior to 7.7.1, the sif file name is WmISTransport_Pre7.7.1.sif. For more information, see " webMethods Transport Files in Siebel Adapter 6.0 SP3" on page 40.

6. Compile your Siebel repository. Make sure you have Write access to your Siebel Server's siebel.srf file.

7. Execute the siebenv.sh file that you edited earlier.

8. Restart your Siebel Server.

Configuring Your System to Use Siebel Visual Basic Scripts

To use Siebel VB scripts, complete the appropriate procedure below. The procedure you choose depends on whether you installed the Siebel Adapter on the same machine as the Siebel Server or on a different machine.

When Siebel Adapter and Siebel Server Are on the Same Machine

To enable Siebel VB scripts when Siebel Adapter and Siebel Server are on the same machine

1. Append jdk\jre\bin\classic to the PATH system environment variable.

2. Append the necessary jar file from the *installation_directory*\common\lib directory to the CLASSPATH system environment variable as follows:
   - For Integration Server 7.1 and later, the jar file is wm-isclient.jar.
   - For Integration Server 7.0 and earlier, the jar file is client.jar.
3. Run the following command:

```
regsvr32
                  Integration Server_directory
                \support\win32\webMethods.dll
```

When Siebel Adapter and Siebel Server Are on Different Machines

To enable Siebel VB scripts when Siebel Adapter and Siebel Server are on different machines

1. On the appropriate Siebel Server machine, create a directory called *drive:*\webMethods, where *drive:* is any valid drive, for example, c:\webMethods.
2. Copy *Integration Server_directory* \support\win32\webMethods.dll to *drive:*\webMethods\webMethods.dll, where *Integration Server_directory* is the directory in which you installed the Integration Server.
3. Copy the necessary jar files as follows:
   - For Integration Server 7.1 and later, copy *installation_directory*\common\lib\wm-isclient.jar to *drive:*\webMethods\lib\wm-isclient.jar.
   - For Integration Server 7.0 and earlier, copy *installation_directory*\lib\client.jar to *drive:*\webMethods\lib\Client.jar.
4. Append *jdk*\jre\bin\classic to the PATH system environment variable, where *jdk* is the directory in which you installed the Java Development Kit.
5. Modify the CLASSPATH statement as follows:
   - For Integration Server 7.1 and later, append *drive*\webMethods\lib\wm-isclient.jar to the CLASSPATH system environment variable.
   - For Integration Server 7.0 and earlier, append *drive*\webMethods\lib\Client.jar to the CLASSPATH system environment variable.
6. Run the following command:

```
regsvr32 drive:\webMethods\webMethods.dll
```

Upgrading to Siebel Adapter 6.0 SP3

You can upgrade from webMethods Enterprise Adapter: Siebel Edition 4.2.1 to Siebel Adapter 6.0 SP3. The upgrade procedure converts operations you configured using the earlier adapter into adapter services that can run on Siebel Adapter 6.0.

The *webMethods Pre-6.0 Enterprise to webMethods 6 Upgrade Guide* describes how to upgrade the earlier adapter and other pre-6.0 webMethods Enterprise business logic components so you can use them in a webMethods 6 development environment. This section explains how to install the upgrade utility software you will need to perform the upgrade process.

To upgrade from webMethods Enterprise Adapter: Siebel Edition 4.2.1

1. Start Software AG Installer 6.1.

2. Go to **Upgrade Utilities > Enterprise Upgrade Utility** and select all the child nodes. The Upgrade Utility installs upgrade tools in the *installation_directory*\Upgrade directory.

3. Select **Adapters > Adapter Upgrades > Enterprise Adapters > webMethods Enterprise Adapter: Siebel Edition 4.2.1 Upgrade Utility Plug-in 6.0**. The plug-in installs the siebelAdapterPlugin.jar file in the *Integration Server_directory* \Upgrade\lib\plugin directory.

4. Perform the upgrade process described in the *webMethods Pre-6.0 Enterprise to webMethods 6 Upgrade Guide*.

Mappings of Configured Operations to Adapter Services

The following table maps the configured operations of the webMethods Enterprise Adapter: Siebel Edition 4.2.1 to the adapter services of the webMethods Siebel Adapter 6.0.

Enterprise Siebel Adapter Configured Operation	Siebel Adapter 6.0 Service
Query	Query
Insert	Insert
Update	Update
Delete	Delete
Associate	Associate
Attachment	File Attachment
Siebel Business Service	Invoke Siebel Business Service Invoke Business Component Method

The Upgrade Utility converts your configured operations to adapter services as completely as possible. However, due to architectural differences between the adapters, the conversion may produce some differences that will not affect the adapter services' run-time functionality.

The Upgrade Utility produces a Manual Tasks folder in the output project that contains tasks you need to perform to complete the upgrade. To locate these manual tasks, drill down through the Manual Tasks folder, then click each task. The Upgrade Utility displays the task instructions. You can also access the manual task instructions in the form of HTML files located in the *UpgradeUtility_directory**project_output*\reports directory. For details, see the *webMethods Pre-6.0 Enterprise to webMethods 6 Upgrade Guide*.

Uninstalling Siebel Adapter 6.0 SP3

To uninstall Siebel Adapter 6.0 SP3

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. Software AG Uninstaller will not delete any user-defined Siebel Adapter 6.0 SP3 components such as connections and adapter services. Because these components will not work without the adapter, delete them manually, either at the file system level or using Digital Event Services or Designer. For instructions, see the *webMethods Service Development Help* for your release.

3. Start Software AG Uninstaller, selecting the webMethods installation directory that contains the host Integration Server. In the product selection list, select **Adapters > webMethods Siebel Adapter 6.0 > Service Pack**. You can also choose to uninstall documentation.

4. Restart the host Integration Server.

Uninstaller removes all Siebel Adapter-related files that were installed. However, Uninstaller does not delete files created after you installed the Siebel Adapter (for example, user-created or configuration files), nor does it delete the directory structure that contains the files. You can go to the *Integration Server_directory* \packages directory and delete the WmSiebelAdapter directory.

3 Package Management

Overview

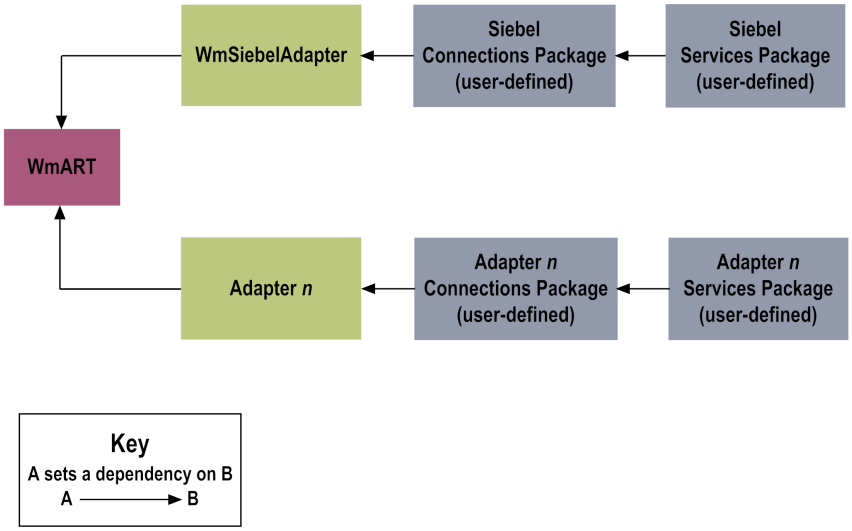
The following sections describe how to set up and manage your Siebel Adapter packages, set up Access Control Lists (ACLs), and use the adapter in a clustered environment.

Siebel Adapter Package Management

The Siebel Adapter is provided as a package called WmSiebelAdapter. You manage this package as you would manage any package on the Integration Server.

When you create connections and adapter services, define them in user-defined packages rather than in the WmSiebelAdapter package. Doing this will allow you to manage the packages more easily, especially when you need to upgrade a deployed adapter. You may create all the connections and adapter services of an adapter in one package, or you may distribute them among multiple packages. For instructions for creating packages, see the *webMethods Service Development Help* for your release.

As you create user-defined packages, use the package management functionality provided in Digital Event Services or Designer and set the user-defined packages to have a dependency on the WmSiebelAdapter package. That way, when the WmSiebelAdapter 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 53).
- "Enabling a Package" on page 54.
- "Disabling a Package" on page 54.

Package Dependency Requirements and Guidelines

Dependency requirements and guidelines for user-defined packages are as follows. For instructions for setting package dependencies, see the *webMethods Service Development Help* for your release.

- By default, the WmSiebelAdapter package has a dependency on the WmART and the WmWin32 packages. Do not change this dependency.
- A user-defined package must have a dependency on the WmSiebelAdapter package.

  These dependencies ensure that at startup Integration Server automatically loads or reloads all packages in the proper order: the WmART package first, the adapter package next, and the user-defined package(s) last. The WmART and the WmWin32 packages are automatically installed when you install the Integration Server. You do not need to manually reload the WmART or the WmWin32 packages.
- If the connections and adapter services of an adapter are defined in different packages, then:
  - A package that contains the connection(s) must have a dependency on the adapter package.
  - Packages that contain adapter services must depend 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.
- 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 a Package" on page 54.
- 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. For information about disabling packages, see "Disabling a Package" on page 54.
- You can name connections and adapter services the same name provided that they are in different folders and packages.

If user-defined packages are properly configured with a dependency on the adapter package, at startup Integration Server automatically loads or reloads all packages in the proper order as follows:

1. The WmART and the WmWin32 packages
2. The WmSiebelAdapter package
3. Your user-defined package(s)

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 Integration Server from loading that package at startup.

Enabling a Package

To enable a package

1. Open Integration Server Administrator.
2. In the **Packages** menu of the navigation area, click **Management**.
3. 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.

> **Important:**
> Before you manually enable a user-defined package, you must first enable its associated adapter package, WmSiebelAdapter. 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.

Disabling a Package

To disable a package

1. Open Integration Server Administrator.
2. In the **Packages** menu of the navigation area, click **Management**.
3. 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 disable the package. When the package is disabled, the server displays **No** in the **Enabled** column.

A disabled adapter package will:

- Remain disabled until you explicitly enable it using the IS Administrator.
- Not appear in Digital Event Services or Designer.

Group Access Control

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 the *webMethods Service Development Help* for your release.

Siebel 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. Because this activity is transparent to the client, clustering makes multiple servers look and behave as one.

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

Integration Server 8.2 SP2 and higher supports the caching and clustering functionality provided by Terracotta. Caching and clustering are configured at the Integration Server level and Siebel Adapter uses the caching mechanism that is enabled on Integration Server.

With clustering, you get the following benefits:

- **Load balancing.** This feature, provided automatically when you set up a clustered environment, allows you to spread the workload over several servers, thus improving performance and scalability.
- **Failover support.** Clustering enables you to avoid a single point of failure. If a server cannot handle a request, or becomes unavailable, the request is automatically redirected to another server in the cluster.

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

- **Scalability.** You can increase your capacity even further by adding new machines running Integration Server to the cluster.

Configuring the Siebel Adapter in a Clustered Environment

When you configure the Siebel 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 56).

- Disable the redirection capability for certain predefined administrative services (see "Disabling the Redirection of Administrative Services" on page 56).

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 Siebel Adapter; that is, you should replicate the Siebel Adapter services and the connections they use.

To ensure consistency, we recommend that 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 *webMethods Integration Server Administrator's Guide* for your release.

Disabling the Redirection of Administrative Services

As mentioned in "What is webMethods Integration Server Clustering?" on page 55, a server that cannot handle a client's service request can automatically redirect the request to another server in the cluster. However, the Siebel Adapter uses certain predefined administrative services that you should not allow to be redirected. These services are used internally when you configure the adapter. If you allow these services to be redirected, your configuration specifications might be saved on multiple servers, which is an undesirable result. For example, if you create two Siebel Adapter services, one might be stored on one server, while the other one might be stored on another server. Remember that all adapter services must reside on all webMethods Integration Servers in the cluster.

To disable the redirection of administrative services

1. Shut down the Integration Server Administrator. For instructions, see the *webMethods Integration Server Administrator's Guide* your release.

2. Edit the following file:

```
  Integration Server_directory
\config\redir.cnf
```

3. Add the following line to the file:

```
<value name="wm.art">false</value>
```

4. Save the file and restart Integration Server.

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* for your release.

Requirements for Each Integration Server in a Cluster

The requirements of each Integration Server in a given cluster are as follows:

- All Integration Servers in a cluster must be of the same version.
- All Siebel Adapters in a cluster must be of the same version.
- All adapter packages on one Integration Server should be replicated to all other Integration Servers in the cluster.
- An identical adapter connection must appear on all servers in the cluster so that any Integration Server in the cluster can handle a given request identically. If you plan to use connection pools in a clustered environment, see "Considerations When Configuring Connections with Connection Pooling Enabled" on page 57.
- Each 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.

See "Replicating Packages to webMethods Integration Servers" on page 56 for information about replicating adapter packages, connections, and adapter services across multiple Integration Servers in a cluster.

Considerations When Installing Siebel Adapter Packages

For each Integration Server in the cluster, use the standard Siebel Adapter installation procedures for each machine, as described in "Installing, Upgrading, and Uninstalling the Siebel Adapter " on page 37.

Considerations When Configuring Connections with Connection Pooling Enabled

When you configure a connection that uses connection pools in a clustered environment, be sure that you do not exceed the total number of connections that can be opened simultaneously for the Siebel application.

For example, if you have a cluster of two Integration Servers with a connection configured to a Siebel application that supports a maximum of 100 connections opened simultaneously, the total number of connections possible at one time must not exceed 100. This means that you cannot

configure a connection with a minimum pool size of 100 and replicate the connection to both servers, because there could be possibly a total of 200 connections opened simultaneously to this Siebel application. You should set the minimum pool size to 0.

In another example, consider a connection configured with a minimum pool size of 10 and a maximum pool size of 100. If you replicate this connection across a cluster with two Integration Servers, it is possible for the connection pool size on both servers to exceed the maximum number of Siebel application connections that can be open at one time.

For information about creating connections for the Siebel Adapter, see " Siebel Adapter Connections" on page 59.

For more general information about connection pools, see the *webMethods Integration Server Administrator's Guide* for your release.

4 Siebel Adapter Connections

Overview

The following sections provide instructions for creating and managing Siebel Adapter connections.

For more information about how connections work, see "Adapter Connections" on page 13.

Before Configuring or Managing Adapter Connections

To prepare to configure or manage an adapter connection

1. Install webMethods Integration Server and Siebel Adapter on the same machine. For details, see "Installing, Upgrading, and Uninstalling the Siebel Adapter " on page 37.

2. Make sure you have webMethods administrator privileges so that you can access the Siebel Adapter administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide* for your release.

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

4. Using Integration Server Administrator, make sure the WmSiebelAdapter package is enabled. See "Enabling a Package" on page 54 for instructions.

5. Using Digital Event Services or Designer, create a user-defined package to contain the connection, if you have not already done so. For more information about managing packages, see " Siebel Adapter Package Management" on page 52.

Run-Time Behavior of Connection Pools in Production Environments

As you configure and test your adapter, set the **Minimum Pool Size** field to **1** or greater. Doing this ensures that Integration Server will verify your connection parameters when you enable (establish) the connection, using the Siebel Adapter.

However, when you deploy the adapter in a production environment, set the **Minimum Pool Size** field to **0** to avoid possible connectivity failures. When the **Minimum Pool Size** field is set to **1** or greater, and the Siebel Server closes an inactive Siebel connection in its pool, the Siebel Server does not notify the Siebel Adapter that this connection has been closed because the Siebel Server has no knowledge of the Siebel Adapter connection pool. If one of your adapter services uses a connection that the Siebel Servers has closed, that adapter service will fail and you will receive a Siebel disconnect message. To eliminate this potential problem, set the **Minimum Pool Size** field to **0**. In this case, instead of keeping one or more connections open at all times (risking the possibility that the Siebel Server may close its inactive connections), the Siebel Adapter connection pool opens one or more connections as needed, eliminating this possibility. Because the pool reuses connections, performance might not be impacted at all. Remember that the connection pool keeps active all connections that have not exceeded the setting specified for the **Expire Timeout** field. For example, if you set **Expire Timeout** to 30000 milliseconds, the pool will retain any inactive connection in

the pool for 30 seconds before closing it. Thus, if an adapter service executes at least once every 30 seconds, the same connection will be reused.

Note:
The **Expire Timeout** interval that you set for the adapter connection must be less than the interval that Siebel's connection pool manager sets for its own connections. For example, if Siebel's connection pool manager sets the interval at 20 seconds and you set the **Expire Timeout** interval at 30 seconds, then the Siebel connection pool manager will always timeout the connections before the adapter's connection pool does.

Configuring Adapter Connections

When you configure a Siebel Adapter connection, you specify information that the adapter uses to connect to and log into a Siebel system. You configure connections using Integration Server Administrator.

As mentioned in the installation chapter, to use a Siebel Java Data Bean client connection, you must obtain the following .jar files from the Siebel Server installation CD and place them in the WmSiebelAdapter\code\jars directory:

- SiebelJI_Common.jar (Not required for Siebel Client version 7.7.1 and later)
- Siebel.jar (Required only for Siebel Client version 7.7.1 and later)
- SiebelJI_*xxx*.jar

  where *xxx* is your locale. For example, SiebelJI_enu.jar for English.

To configure a connection

1. In Integration Server Administrator, select **Adapters > Siebel Adapter**.
2. On the Connections screen, click **Configure New Connection**.
3. On the Connection Types screen, select one of the following adapter connection types:

Connection Type	Description
Siebel Java Connection	Uses Siebel's Java Data Bean Business Object Interface (BOI) API to communicate with the Siebel Server. This connection requires that your Windows or UNIX platform supports a JVM that the Data Bean supports. When configuring a connection that will connect to a virtual server used for Siebel Server load balancing, select this connection type. For more information about this feature, see "Using Connections with Siebel Server Load Balancing" on page 14.

Connection Type	Description
Siebel Windows Thin Connection	Uses Siebel's COM Data Control to communicate with the Siebel Server's Object Manager.

The Configure Connection Type screen is displayed. For more information, see "Connection Types" on page 13.

4. In the **Configure Connection Type** section, provide values for the following parameters:

Parameter	Description/Action
Package	The package in which to create the connection. You must create the package using Digital Event Services or Designer before you can specify it using this parameter. For general information about creating packages, see the *webMethods Service Development Help* for your release. **Note:** Create the connection in a user-defined package rather than in the adapter's package. See " Siebel Adapter Package Management" on page 52 for other important considerations when creating packages for the Siebel Adapter.
Folder Name	The folder in which to create the connection.
Connection Name	The name you want to give the connection (32 characters maximum). Connection names cannot have spaces or use special characters reserved by Integration Server, Digital Event Services or Designer. For more information about the use of special characters in package, folder, and element names, see the *webMethods Service Development Help* for your release.

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

Parameter	Description/Action
Transport	Optional. The protocol to communicate with the Siebel Server. Type **tcpip** (the default) or **http**.
Encryption	Optional. The ability to encrypt data. Select **none** (the default) or **mscrypto**.
Compression	Optional. The data in compressed format. Select **none** (the default), **zlib**, or **pkware**.

Parameter	Description/Action
Gateway Server	The name of the Siebel Gateway Server through which you want to open a session to the Siebel Server. Enter the server name or the server's IP address. **Note:**If this connection will connect to a virtual server used for Siebel Server load balancing, enter the name of the Siebel virtual server for this field. For more information about this feature, see "Using Connections with Siebel Server Load Balancing" on page 14.
Siebel Enterprise	The name of the Siebel Enterprise to which you want to connect.
Object Manager	The name of the Siebel Object Manager through which you want to access the Siebel application's business objects.
Siebel Server	The name of the Siebel Server through which you want to access the Siebel application's business objects. You do not need to provide the name of the Siebel Server for Siebel Server version 7.7.1 and later. **Note:**If this connection will connect to a virtual server used for Siebel Server load balancing, leave this field blank. For more information about this feature, see "Using Connections with Siebel Server Load Balancing" on page 14.
Repository Context	The Siebel repository in which the Siebel application stores the business component and business object definitions. **Note:** Specify the same repository that your Siebel Server uses.
Language	A valid Siebel language code specifying which language the Siebel connection uses. For example, ENU specifies English. For a list of all available codes, see your Siebel documentation.
User Name	The login name to use to log in to the Siebel Server. If no login is required, leave this field blank.
Password	The password to use to log in to the Siebel Server. If no login is required, leave this field blank.

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

Parameter	Description/Action
Enable Connection Pooling	Enables the adapter to use connection pooling. Default: **true**.

Parameter	Description/Action
	For more information about connection pooling, see "Connection Pooling" on page 14 and "Run-Time Behavior of Connection Pools in Production Environments" on page 60. **Note:** If you plan to enable connection pooling in a clustered environment, consider the connection pool size. For details, see "Clustering Considerations and Requirements" on page 57.
Minimum Pool Size	The minimum number of connection objects that remain in the connection pool at all times, even if the connections become idle. When the adapter creates the pool, it creates this number of connections. Default: **1**. **Important:** As you configure and test your adapter, we recommend that you set the **Minimum Pool Size** field to **1** or greater. Doing this ensures that the adapter will verify your connection parameters when you enable (establish) the connection, using the Siebel Adapter. However, when you deploy the adapter in a production environment, set the **Minimum Pool Size** field to **0** to avoid possible connectivity failures. For details, see "Run-Time Behavior of Connection Pools in Production Environments" on page 60.
Maximum Pool Size	The maximum number of connection objects that can exist at one time 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. Default: **10**.
Pool Increment Size	The number of connections by which the pool will be incremented if connections are needed, up to the maximum pool size. Default: **1**.
Block Timeout	The number of milliseconds that Integration Server will wait to obtain a connection with the Siebel application before it times out and returns an error. Default: **1000** milliseconds.
Expire Timeout	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. Default: **1000** milliseconds. **Note:** The **Expire Timeout** interval that you set for the adapter connection must be less than the interval that Siebel's connection pool manager sets for its own connections. For example, if Siebel's connection pool manager sets the interval at 20 seconds and you set the **Expire**

Parameter	Description/Action
	Timeout interval at 30 seconds, then the Siebel connection pool manager will always timeout the connections before the adapter's connection pool does. **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	The number of times that the system should attempt to initialize the connection pool at startup if the initial attempt fails. Default: **0**.
Startup Backoff Timeout	The number of seconds that the system should wait between attempts to initialize the connection pool.

7. Click **Save Connection**.

   The connection you created appears on the adapter's Connections screen and in the Digital Event Services Service Browser or the Designer Package Navigator.

8. Enable the connection, as described in "Enabling Adapter Connections" on page 69.

   When you enable the connection, Integration Server verifies your connection parameters and tries to establish a connection to the Siebel application, assuming you configured your connection with the **Minimum Pool Size** field set to **1** or greater. If you set the **Minimum Pool Size** field set to **0,** Integration Server does not try to establish a connection until you use the connection to configure adapter services.

Dynamically Changing a Service's Connection at Run Time

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

For example, you have a flow whose primary purpose is to update a business component (for example, Contact) on a production Siebel server. However, you want the flow to have the capability to update the same business component on a test Siebel server-with the decision of which Siebel server to update to be made programmatically at run time. The output signature of the flow's first service contains a field called Target. The flow could branch based on the value in Target. If Target contains the value Production, the second service in the flow would ignore $connectionName-thus using its default connection to connect to (and then update) the production Siebel server. However, if Target contains the value Test, the second service in the flow would use the value in the $connectionName from the pipeline and connect to (and then update) the test Siebel server.

Keep in mind these restrictions when using dynamic connections:

- Both connections-the default and override-must connect to the same version of Siebel Server.
- The $connectionName field is present only in services…
  - created with Digital Event Services 6.5 or later
  - created with Digital Event Services 6.1 (or earlier) and edited with Digital Event Services 6.5
  - created with Designer

For more information, see "Changing the Connection Associated with an Adapter Service at Run Time" on page 20.

Viewing Adapter Connection Parameters

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

Using Integration Server Administrator

To view the parameters for a connection using the Integration Server Administrator

1. In Integration Server Administrator, select **Adapters > Siebel 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 35.

2. On the Connections screen, click the icon 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 61.

3. Click **Return to Siebel Adapter Connections** to return to the main connections screen.

Using Digital Event Services or Designer

To view the parameters for a connection using Digital Event Services or Designer

1. Start Digital Event Services or Designer if it is not already running.
2. From the Digital Event Services Navigation panel or the Designer Package Navigator view, open the package and folder in which the connection is located.
3. Double-click the connection you want to view.
4. The parameters for the connection appear in the **Connection Information** tab. For descriptions of the connection parameters, see "Configuring Adapter Connections" on page 61.

Editing Adapter Connections

If the login information for your Siebel Server changes, or if you want to redefine parameters that a connection uses when connecting to a Siebel Server, you must update your connection. You edit adapter connections using the Integration Server Administrator.

To edit a connection

1. Disable the connection that you want to edit. See "Disabling Adapter Connections" on page 69 for instructions.
2. In Integration Server Administrator, select **Adapters > Siebel Adapter**.
3. On the Connections screen, click the icon for the connection you want to edit.

   The Edit Connection screen displays the current parameters for the connection. Update the connection's parameters by typing or selecting the values you want to specify.

   For descriptions of the connection parameters, see the table of parameters in "Configuring Adapter Connections" on page 61.
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 69 for instructions.

Copying Adapter Connections

You can copy an existing Siebel Adapter connection to configure a new connection with the same or similar connection properties without having to re-type all of the properties for the connection. You copy adapter connections using Integration Server Administrator.

To copy a connection

1. In Integration Server Administrator, select **Adapters > Siebel Adapter**.

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

   The Copy Connection screen displays the current parameters for the connection you want to copy. Name the new connection, specify a Package Name and Folder Name, and edit any connection parameters as needed by typing or selecting the values you want to specify.

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

   For descriptions of the connection parameters, see "Configuring Adapter Connections" on page 61.

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

Deleting Adapter Connections

If you no longer want to use a particular Siebel Adapter connection, you can delete it by following the instructions in this section. You delete adapter connections using Integration Server Administrator.

If you delete a Siebel Adapter connection, the adapter services that are defined to use the connection will no longer work. However, you can change which connection an adapter service uses. Therefore, if you delete a Siebel Adapter connection, you can assign a different connection to an adapter service and re-use the service. To do this, you use the built-in webMethods pub.art.service:setAdapterServiceNodeConnectionservice. For more information, see "Changing the Connection Associated With an Adapter Service at Design Time" on page 19.

To delete a connection

1. Disable the connection you want to delete. See "Disabling Adapter Connections" on page 69 for instructions.

2. In Integration Server Administrator, select **Adapters > Siebel Adapter**.

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

Integration Server deletes the adapter connection.

Enabling Adapter Connections

A Siebel Adapter connection must be enabled before you can configure any adapter service using the connection, or before an adapter service can use the connection at run time. You enable adapter connections using Integration Server Administrator.

When you enable a connection, the adapter verifies your connection parameters and tries to establish a connection to the Siebel application, assuming you configured your connection with the **Minimum Pool Size** field set to **1** or greater. However, when you deploy the adapter in a production environment, set the **Minimum Pool Size** field to **0** to avoid possible connectivity failures. For details, see "Run-Time Behavior of Connection Pools in Production Environments" on page 60.

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

To enable a connection

1. In Integration Server Administrator, select **Adapters > Siebel 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 a ✔ and **Yes** in the **Enabled** column.

Disabling Adapter Connections

Siebel Adapter connections must be disabled before you can edit or delete them. You disable adapter connections using Integration Server Administrator.

To disable a connection

1. In Integration Server Administrator, select **Adapters > Siebel 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 a **No** in the **Enabled** column.

5 webMethods-To-Siebel Communication

Overview

The following sections describe how to configure and manage adapter services, which you use for webMethods-to-Siebel communication. For a description of adapter services, see " webMethods-to-Siebel Communication" on page 18.

The following table lists the tasks required to use adapter services:

Task		Use this tool...
1	Configure an adapter connection. See " Siebel Adapter Connections" on page 59 for details.	Integration Server Administrator
2	Select the appropriate adapter service template and configure the adapter service, as described in the following sections.	Digital Event Services or Designer
3	If you plan to use an Integration Server flow or Java service to invoke the adapter service, design the flow or Java service to use this adapter service. For details, see the *webMethods Service Development Help* for your release.	Digital Event Services or Designer
4	Manage the adapter service. See "Package Management" on page 51 and "Logging and Exception Handling" on page 113 for details.	Digital Event Services or Designer and Integration Server Administrator

Note:
Each service operates on specific fields of a specific Siebel business component. When configuring adapter services, you need to consider the properties (such as default field values) of your Siebel business components. Business components also contain predefined procedures that you can invoke using an adapter service. Therefore, you should be very familiar with your Siebel business components before creating adapter services that operate on them.

Before Configuring or Managing Adapter Services

To prepare to configure or manage an adapter service

1. Start Integration Server and Integration Server Administrator.

2. Make sure you have webMethods Integration Server administrator privileges so that you can access Integration Server Administrator's administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide* your release.

3. Using Integration Server Administrator, make sure the WmSiebelAdapter package is enabled. See "Enabling a Package" on page 54 for instructions.

4. Using Integration Server Administrator, configure the adapter connection you plan to use with the adapter service. For instructions, see "Configuring Adapter Connections" on page 61.

5. Start Digital Event Services or Designer if it is not already running.

   Note:
   If you are using Digital Event Services 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 Service Development Help* for your release.

6. Using Digital Event Services or Designer, create a user-defined package to contain the service, if you have not already done so. When you configure adapter services, you should always define them in user-defined packages rather than in the WmSiebelAdapter package. For more information about managing packages for the adapter, see "Package Management" on page 51.

Configuring Query, Insert, Update, and Delete Services

These services perform the following operations:

Service	Description
Query	Retrieves business component records based on one or more fields.
Insert	Inserts a business component record with values specified for one or more fields.
Update	Updates business component records based on the value of one or more fields. You can specify that one or more record fields be updated.
Delete	Deletes business component records based on the value of one or more fields.

For more information, see "Query Services" on page 21, "Insert Services" on page 22, "Update Services" on page 23, and "Delete Services" on page 24.

Note:
When creating a Query, Insert, Update, or Delete service, you can select a multi-valued link (MVL) to a child record. For more information, see "Multi-Valued Links in Services" on page 19.

To configure a Query, Insert, Update, or Delete service

1. Review the steps in "Before Configuring or Managing Adapter Services" on page 72.

2. Start Digital Event Services or Designer.

   Note:
   Make sure the server with which you want to use Digital Event Services or Designer is running.

3. Perform one of the following:

- If you are using Digital Event Services, 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 **Siebel Adapter** from the list of available adapter types. Click **Next**.
5. Select the appropriate **Adapter Connection Name** and click **Next**.
6. From the list of available templates, select the **Query**, **Insert**, **Update**, or **Delete** template as appropriate and do one of the following:
   - If you are using Digital Event Services, click **Next**, select a package and folder where the adapter service should be stored, and then click **Finish**.
   - If you are using Designer, click **Finish**.
7. Type a unique name for the service and select the appropriate folder. Click **Finish**.
8. You can select the **Adapter Settings** tab at any time to confirm adapter properties such as adapter type, connection name, and service template, as needed.
9. Select the **Navigation Path** tab to specify the records that the service accesses, using the following fields:

Parameter	Description/Action
Business Object	The name of the business object that contains the business component for which you are creating the service.
Business Component	The name of the business component for which you are creating the service.
Multi-Valued Links	Optional. Select a field of the business component to use as a multi-valued link (MVL) to a child record. For example, to query the primary business address of a particular Account, you might select the Account business component's field Company Name, which links to a business component named Business Address. This creates a query that navigates from the Account business component to the Business Address business component. For more information about multi-valued links, see "Multi-Valued Links in Services" on page 19.

Parameter	Description/Action
Additional Business Component [1]	The parent business component of the business component you selected, if one exists.
Additional Business Component [2]	The parent business component of the business component you selected in Additional Business Component [1], if one exists.

10. Use the **Search** tab to define a search expression for the selected business component. To define a search expression, provide values for the fields on the top row of the tab. Also define search expressions for a multi-valued link and up to two additional business components, if you specified them.

    a. Use the icon (or the icon) to create new rows as needed.

    b. Select a logical operator from the **AND/OR** field, an **Operator**, and separators (the left and right parentheses) as needed, and provide values for the following parameters:

Parameter	Description/Action
AND/OR	A logical operator.
Siebel Field	A search field of the selected business component.
Siebel Field Data Type	The data type of **Siebel Field**.
Operator	An operator for the search expression. **Note:** For case insensitive search, select '~='. For case sensitive search, select '='.
Input Field	An input field of the adapter service.
Input Field Data Type	The data type of **Input Field**.

11. If you want to perform a Query service

    a. Select the **Output Mapping** tab to map the selected Siebel fields to output fields of the adapter service.

    b. Provide values for the following **Output Mapping** parameters:

Parameter	Description/Action
Siebel Field	Select a Siebel field to use as output to the adapter service.
Siebel Field Data Type	The data type of **Siebel Field**.

Parameter	Description/Action
Output Field	The field to map **Siebel Field** to.
Output Field Type	The data type of **Output Field**.
Sort Order	Select a sort order (**Ascending** or **Descending**).

12. If you want to perform an Insert or Update service

    a. Select the **Input Mapping** tab to map the adapter service's input fields to Siebel fields.

    b. Provide values for the following **Input Mapping** parameters:

Parameter	Description/Action
Siebel Field	Select a Siebel field to use as input to the adapter service.
Siebel Field Data Type	The data type of **Siebel Field**.
Input Field	The field to map **Siebel Field** to.
Input Field Data Type	The data type of **Output Field**.

Note:
When mapping input values for a Siebel field with a drop-down list, make sure you provide the predefined values provided in the drop-down list. For example, assume that the Status field provides the drop-down list values Open, Pending, and Closed. When you insert or update a record that contains the Status field, the Siebel application expects you to provide Open, Pending, or Closed. Otherwise, the adapter throws an exception.

13. For information about using the **Input/Output**, **Settings**, **Audit**, **Permissions**, and **Results** tabs, see the *webMethods Service Development Help* for your release.

    Digital Event Services and Designer contain the **Adapter Settings** and **Input/Output** tabs. The information from the **Audit** and **Permissions** tabs now appears in the **Properties** panel, and the information from the **Results** tab now appears in the **Results** panel.

14. Select **File > Save**.

15. To run the adapter service directly from Digital Event Services or Designer at this time, perform the following steps:

    a. If you are using Digital Event Services, set the view to the Test perspective.

b. In Digital Event Services or Designer, expand the package and folder that contain the service you want to test.

c. Double-click the service you want to test.

   Digital Event Services or Designer displays the configured service in the service template's Adapter Service Editor.

d. Perform one of the following

   - If you are using Digital Event Services, select **Test > Run**.
   - If you are using Designer, select **Run > Run As > Run Service**.

e. To connect to the Siebel Server using credentials that are different than the credentials provided at design time, specify the username and password here:

Parameter	Description/Action
user	The user name used to connect to the Siebel Server.
password	The password used to connect to the Siebel Server.

Note:
If you do not provide the user credentials here, the user credentials specified when the connection was configured are used. For more information, see "Configuring Adapter Connections" on page 61 and "Passing Credentials at Run Time while Invoking an Adapter Service" on page 20.

f. For every service input field, you will be prompted to enter an input value. Enter a value for each input field and then click **OK**.

g. Click the **Results** tab to view the output from this service.

   The output for each service is as follows:

The output for this service ...	Is a ...
Query	Document list field named Results. If the Query was successful, a document field (also named Results) displays the fields returned by the service.
Update	String field named Rows Updated, which indicates the number of rows updated by the service.
Insert	String field named Row ID, which indicates the row ID of the row inserted by the service.

The output for this service ...	Is a ...
Delete	String field named Rows Deleted, which indicates the number of rows deleted by the service.

Configuring Associate Services

The Associate service performs a Siebel Associate operation. This service establishes relationships among multi-value group (MVG) business component records.

For more information, see "Associate Services" on page 25.

To configure an Associate service

1. Review the steps in "Before Configuring or Managing Adapter Services" on page 72.
2. Start Digital Event Services or Designer.

   Note:
   Make sure the server with which you want to use Digital Event Services or Designer is running.

3. Perform one of the following:
   - If you are using Digital Event Services, 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 **Siebel Adapter** as the adapter type and click **Next**.
5. Select the appropriate **Adapter Connection Name** and click **Next**.
6. Select the **Associate** template and click **Next**.
   - If you are using Digital Event Services, 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**.

7. The service is created and its parameters and controls are displayed in the adapter service editor. You can select the **Adapter Settings** tab at any time to confirm adapter properties such as adapter type, connection name, and service template, as needed.

8. Select the **Navigation Path** tab to specify the records that the service accesses, using the following fields:

Parameter	Description/Action
Business Object	The name of the business object that contains the business component for which you are creating a service.
Business Component	The name of the business component for which you are creating the service.
Multi-Valued Links	Select a field of the business component to use as a multi-valued link (MVL) to the business component that you want to associate. For example, if you selected a business component named Orders, you might want to associate with it all the order line items. That is, you might select a field named Line Item, which is a multi-valued link to a business component named Order Line Item. You can associate only those component records that are linked by an association relationship. These component records are identified by the suffix *ASSOCIATE. Fields without this suffix are not linked by an association; they represent a one-to-many relationship, rather than a many-to-many relationship.
Additional Business Component [1]	The parent business component of the business component you selected, if one exists.
Additional Business Component [2]	The parent business component of the business component you selected in Additional Business Component [1], if one exists.

9. Use the **Search** tab to define a search expression for the selected business component. To define a search expression, provide values for the fields on the top row of the tab. Also define search expressions for a multi-valued link and up to two additional business components, if you specified them.

   a. Use the icon (or the icon) to create new rows as needed.

   b. Select a logical operator from the **AND/OR** field, an **Operator**, and separators (the left and right parentheses) as needed, and provide values for the following parameters:

Parameter	Description/Action
AND/OR	A logical operator.
Siebel Field	A search field of the selected business component.

Parameter	Description/Action
Siebel Field Data Type	The data type of **Siebel Field**.
Operator	An operator for the search expression. **Note:** For case insensitive search, select '~='. For case sensitive search, select '='.
Input Field	An input field of the adapter service.
Input Field Data Type	The data type of **Input Field**.

10. For information about using the **Input/Output**, **Settings, Audit**, **Permissions**, and **Results** tabs, see the *webMethods Service Development Help* for your release.

    Digital Event Services and Designer contain the **Adapter Settings** and **Input/Output** tabs. The information from the **Audit** and **Permissions** tabs now appears in the **Properties** panel, and the information from the **Results** tab now appears in the **Results** panel.

11. Select **File > Save**.

12. To run the adapter service directly from Digital Event Services or Designer at this time, perform the following steps:

    a. If you are using Digital Event Services, set the view to the Test perspective.

    b. In Digital Event Services or Designer, expand the package and folder that contain the service you want to test.

    c. Double-click the service you want to test.

       Digital Event Services or Designer displays the configured service in the service template's Adapter Service Editor.

    d. Perform one of the following

       - If you are using Digital Event Services, select **Test > Run**.
       - If you are using Designer, select **Run > Run As > Run Service**.

    e. To connect to the Siebel Server using credentials that are different than the credentials provided at design time, specify the username and password here:

Parameter	Description/Action
user	The user name used to connect to the Siebel Server.

Parameter	Description/Action
password	The password used to connect to the Siebel Server.

Note:
If you do not provide the user credentials here, the user credentials specified when the connection was configured are used. For more information, see "Configuring Adapter Connections" on page 61 and "Passing Credentials at Run Time while Invoking an Adapter Service" on page 20.

f. If you defined any variable input fields, you will be prompted to enter their input values. Click **OK**.

g. Click the **Results** tab to view the output from this service. The output is a string field named Rows Associated, which indicates the number of rows of the associated MVG business component record.

Configuring Attachment Services

An Attachment service creates an attachment file in a business component record, updates an existing attachment, or obtains an attachment. To do this, at run time you specify that the Attachment service invokes one of the following Siebel methods: CreateFile, PutFile, or GetFile, respectively.

For more information, see "Attachment Services" on page 26.

To configure an Attachment service

1. Review the steps in "Before Configuring or Managing Adapter Services" on page 72.

2. Start Digital Event Services or Designer.

   Note:
   Make sure the server with which you want to use Digital Event Services or Designer is running.

3. Perform one of the following:

   - If you are using Digital Event Services, 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 **Siebel Adapter** as the adapter type and click **Next**.

5. Select the appropriate **Adapter Connection Name** and click **Next**.

6. Select the **Attachment** template and click **Next**.

   - If you are using Digital Event Services, 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**.

7. The service is created and its parameters and controls are displayed in the adapter service editor. You can select the **Adapter Settings** tab at any time to confirm adapter properties such as adapter type, connection name, and service template, as needed.

8. Select the **Navigation Path** tab to specify the records that the service accesses, using the following fields:

Parameter	Description/Action
Business Object	The name of the business object that contains the business component for which you are creating the service. For example, Contact.
Business Component	The name of the business component for which you are creating the service. For example, Contact Attachment.
Multi-Valued Links	Leave blank.
Additional Business Component [1]	Leave blank.
Additional Business Component [2]	Leave blank.

9. If you plan to use the Attachment service to execute a GetFile or PutFile method, select the **Search** tab and define a search expression for the selected business component by providing values for the fields on the top row of the tab. Note that if you plan to use the Attachment service to execute a CreateFile method, leave the **Search** tab blank.

   a. Use the icon (or the icon) to create new rows as needed.

   b. Select a logical operator from the **AND/OR** field, an **Operator**, and separators (the left and right parentheses) as needed, and provide values for the following parameters:

Parameter	Description/Action
AND/OR	A logical operator.
Siebel Field	A search field of the selected business component.
Siebel Field Data Type	The data type of **Siebel Field**.
Operator	An operator for the search expression. **Note:** For case insensitive search, select '~='. For case sensitive search, select '='.
Input Field	An input field of the adapter service to which you will map the corresponding value in **Siebel Field**. The adapter automatically populates this field with values identical to the values in **Siebel Field**.
Input Field Data Type	The data type of **Input Field**.

For example, to search on a business component's ID field and map its value to an input field named ID, you would specify:

ID DTYPE_ID = ID java.lang.String

The ID field is the Siebel field Row-id.

10. The **Attachment Operation Inputs** tab is a display-only tab. You will provide values for the fields on this tab when you test the service, as described in "Testing the Attachment Service" on page 83.

11. For information about using the **Input/Output**, **Settings, Audit**, **Permissions**, and **Results** tabs, see the *webMethods Service Development Help* for your release.

    Digital Event Services and Designer contain the **Adapter Settings** and **Input/Output** tabs. The information from the **Audit** and **Permissions** tabs now appears in the **Properties** panel, and the information from the **Results** tab now appears in the **Results** panel.

12. Select **File > Save**.

Testing the Attachment Service

To test an Attachment service, you can run it from either Digital Event Services or Designer before you run it in a flow or Java service. You can run any Attachment service to either create, update, or obtain an attachment file. To do this, at run time you specify that the Attachment service invokes one of the following Siebel methods:

Siebel Method	Description
CreateFile	Creates a file in the Siebel file system from an external source.
GetFile	Obtains a file from the Siebel file system and places that file on the local file system of the Siebel Server.
PutFile	Updates a file in the Siebel file system with the version of the file on the client machine.

The following procedure describes how to test an Attachment service by running it directly from Digital Event Services or Designer. The procedure provides example values for an Attachment service that invokes the CreateFile method for a business object named Contact and a business component named ContactAttachment.

To test an Attachment service from Digital Event Services or Designer

1. If you are using Digital Event Services, set the view to the Test perspective.

2. In Digital Event Services or Designer, expand the package and folder that contain the service you want to test.

3. Double-click the service you want to test.

   Digital Event Services or Designer displays the configured service in the service template's Adapter Service Editor.

4. Perform one of the following

   - If you are using Digital Event Services, select **Test > Run**.
   - If you are using Designer, select **Run > Run As > Run Service**.

5. To connect to the Siebel Server using credentials that are different than the credentials provided at design time, specify the username and password here.

Parameter	Description/Action
user	The user name used to connect to the Siebel Server.
password	The password used to connect to the Siebel Server.

Note:
If you do not provide the user credentials here, the user credentials specified when the connection was configured are used. For more information, see "Configuring Adapter Connections" on page 61 and "Passing Credentials at Run Time while Invoking an Adapter Service" on page 20.

6. A pop-up dialog box appears, with input parameters for the service. Complete the dialog box appropriately for the Siebel method you want the Attachment service to invoke, as follows:

   - To invoke the Siebel method CreateFile, complete the dialog box as follows:

CreateFile Input Parameter	Description/Action
Method_To_Invoke	Specify CreateFile.
Keep_Link	Applies to URLs. Specify Y (Yes) or N (No) depending on whether a link to the file is stored as an attachment instead of the actual file. The example specifies Y.
Path_To_Source_File	Specify the fully qualified path of the file. The file can be located on the Siebel Server or on another computer that the Siebel Server can access.
Key_Field_Name	Specify the name of the field in the business component that contains the file name. In the example, this is ContactFileName, which is a Siebel field in the ContactAttachment business component.
Name_Of_ID_Field	Specify the ID field of the attachment business component. In the example, this is the field Contact Id of the business component ContactAttachment.
ID	The Siebel Row-id value of the attachment you are creating. In the example, this is the Row # of the contact to which this attachment is assigned.

   - To invoke the Siebel method GetFile, complete the dialog box as follows:

GetFile Input Parameter	Description/Action
Method_To_Invoke	Specify GetFile.
Keep_Link	Leave blank.
Path_To_Source_File	Leave blank.
Key_Field_Name	Specify the name of the field in the business component that contains the file name. In the example, you would specify ContactFileName, which is a Siebel field in the ContactAttachment business component.
Name_Of_ID_Field	Leave blank.
ID	The Siebel Row-id value of the attachment you are retrieving.

- To invoke the Siebel method PutFile, complete the dialog box as follows:

PutFile Input Parameter	Description/Action
Method_To_Invoke	Specify PutFile.
Keep_Link	Leave blank.
Path_To_Source_File	Specify the fully qualified path of the file on the Siebel Server.
Key_Field_Name	Specify the name of the field in the business component that contains the file name. In the example, you would specify ContactFileName, which is a Siebel field in the ContactAttachment business component.
Name_Of_ID_Field	Leave blank.
ID	The Siebel Row-id value of the attachment you are updating.

7. Click **OK**.

8. Click the **Results** tab to view the output from this service. The output is a string field named Status, which indicates Success or Failure. If the service invoked a GetFile method, the field also indicates the path of the attachment returned by the service.

Note:
To pass empty values for particular fields, leave those fields blank and select the option **Include empty values for String Types**.

Configuring Services That Invoke Siebel Business Services

The Business Service service invokes a Siebel business service on your Siebel Server. For more information, see "Business Service Services" on page 28.

To configure a service that invokes a Siebel business service

1. Review the steps in "Before Configuring or Managing Adapter Services" on page 72.

2. Start Digital Event Services or Designer.

   Note:
   Make sure the server with which you want to use Digital Event Services or Designer is running.

3. Perform one of the following:

- If you are using Digital Event Services, 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 **Siebel Adapter** as the adapter type and click **Next**.

5. Select the appropriate **Adapter Connection Name** and click **Next**.

6. Select the **BusinessService** template and click **Next**.

   - If you are using Digital Event Services, 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**.

7. You can select the **Adapter Settings** tab at any time to confirm adapter properties such as adapter type, connection name, and service template, as needed.

8. Select the **Siebel Business Service Inputs/Outputs** tab to configure the inputs and outputs of the Siebel business service that the adapter service invokes, as follows:

   a. Use the icon (or the icon) to create new rows as needed.

   b. Provide values for the following parameters:

Parameter	Description/Action
Business Service Inputs	Specify the: ■ Name of the business service. ■ Name of the business service method. ■ Array of Siebel property set keys. ■ Array of Siebel property set values.
Business Service Property Output Set - Elements	Specify the: ■ Array of Siebel property set keys. ■ Array of Siebel property set values. ■ Status of the business service execution.

9. For information about using the **Input/Output**, **Settings, Audit**, **Permissions**, and **Results** tabs, see the *webMethods Service Development Help* for your release.

   Digital Event Services and Designer only contain the **Adapter Settings** and **Input/Output** tabs. The information from the **Audit** and **Permissions** tabs now appears in the **Properties** panel, and the information from the **Results** tab now appears in the **Results** panel.

10. Select **File > Save**.

11. To run the adapter service directly from Digital Event Services or Designer at this time, perform the following steps:

    a. If you are using Digital Event Services, set the view to the Test perspective.

    b. In Digital Event Services or Designer, expand the package and folder that contain the service you want to test.

    c. Double-click the service you want to test.

       Digital Event Services or Designer displays the configured service in the service template's Adapter Service Editor.

    d. Perform one of the following

       - If you are using Digital Event Services, select **Test > Run**.
       - If you are using Designer, select **Run > Run As > Run Service**.

    e. To connect to the Siebel Server using credentials that are different than the credentials provided at design time, specify the username and password here:

Parameter	Description/Action
user	The user name used to connect to the Siebel Server.
password	The password used to connect to the Siebel Server.

Note:
If you do not provide the user credentials here, the user credentials specified when the connection was configured are used. For more information, see "Configuring Adapter Connections" on page 61 and "Passing Credentials at Run Time while Invoking an Adapter Service" on page 20.

    f. If you defined any variable input fields, you will be prompted to enter their input values. Click **OK**.

    g. Click the **Results** tab to view the output (the result of the Siebel business service execution) from this service.

Configuring Services That Invoke Business Component Methods

The Invoke Business Component Method service invokes a Siebel business component method on your Siebel Server. For more information, see "Invoke Business Component Method Services" on page 29.

To configure an Invoke Business Component Method service

1. Review the steps in "Before Configuring or Managing Adapter Services" on page 72.

2. Start Digital Event Services or Designer.

   Note:
   Make sure the server with which you want to use Digital Event Services or Designer is running.

3. Perform one of the following:

   - If you are using Digital Event Services, select **File > New > Adapter Service** and click **Next**.
   - If you are using Designer, right-click the package in which the service should be contained and select **New > Adapter Service**.

4. Select the **InvokeBusCompMethod** template and click **Next**.

   - If you are using Digital Event Services
     1. Click **Next**.
     2. Select a package and folder to contain the service.
     3. Type a unique name for the service.
     4. Click **Finish**.
   - If you are using Designer, click **Finish**.

5. You can select the **Adapter Settings** tab at any time to confirm adapter properties such as adapter type, connection name, and service template, as needed.

6. Select the **Navigation Path** tab to specify the records that the service accesses, using the following fields:

Parameter	Description/Action
Business Object	The name of the business object that contains the business component for which you are creating the service.
Business Component	The name of the business component whose method you want to invoke.
Multi-Valued Links	Optional. Select a field of the business component to use as a multi-valued link (MVL) to a child record. For example, to query the primary business address of a particular Account, you might select the Account business component's field Company Name, which links to a business component named Business Address. This creates a query that navigates from the Account business component to the Business Address business component. For more information, see "Multi-Valued Links in Services" on page 19.
Additional Business Component [1]	The parent business component of the business component you selected, if one exists.
Additional Business Component [2]	The parent business component of the business component you selected in Additional Business Component [1], if one exists.

7. The **Invoke Business Component Method - Inputs** tab is a display-only tab, with the input fields **Business_Component_Method_Name** and **String_Array_of_Input_Arguments**. You will provide values for these fields to run the service, as described in "Testing the Invoke Business Component Method Service" on page 90.

8. For information about using the **Input/Output**, **Settings, Audit**, **Permissions**, and **Results** tabs, see the *webMethods Service Development Help* for your release.

   Digital Event Services and Designer only contain the **Adapter Settings** and **Input/Output** tabs. The information from the **Audit** and **Permissions** tabs now appears in the **Properties** panel, and the information from the **Results** tab now appears in the **Results** panel.

9. Select **File > Save**.

Testing the Invoke Business Component Method Service

To test an Invoke Business Component Method service, you can run it from either Digital Event Services or Designer before you run it in a flow or Java service.

To test an Invoke Business Component Method service from Digital Event Services or Designer

1. If you are using Digital Event Services, set the view to the Test perspective.

2. In Digital Event Services or Designer, expand the package and folder that contain the service you want to test.

3. Double-click the service you want to test.

   Digital Event Services or Designer displays the configured service in the service template's Adapter Service Editor.

4. Perform one of the following

   - If you are using Digital Event Services, select **Test > Run**.
   - If you are using Designer, select **Run > Run As > Run Service**.

5. To connect to the Siebel Server using credentials that are different than the credentials provided at design time, specify the username and password here:

Parameter	Description/Action
user	The user name used to connect to the Siebel Server.
password	The password used to connect to the Siebel Server.

Note:
If you do not provide the user credentials here, the user credentials specified when the connection was configured are used. For more information, see "Configuring Adapter Connections" on page 61 and "Passing Credentials at Run Time while Invoking an Adapter Service" on page 20.

6. A pop-up dialog box appears, with input parameters for the service.

7. Type the name of the method to invoke in the **Business_Component_Method_Name** field.

8. Use the icon to create a new row in the **String_Array_of_Input_Arguments** field, and type the arguments for the method.

9. Click **OK**.

10. Click the **Results** tab to view the output from this service. The output is a string field named Status, which contains the output generated by the method that is invoked.

Note:
To pass empty values for particular fields, leave those fields blank and select the option **Include empty values for String Types**.

Testing Adapter Services

You use Digital Event Services or Designer to test adapter services.

For more information about testing and debugging services, see the *webMethods Service Development Help* for your release.

To test adapter services

1. Review the steps in "Before Configuring or Managing Adapter Services" on page 72.
2. If you are using Digital Event Services, set the view to the Test perspective.
3. In Digital Event Services or Designer, expand the package and folder that contain the service you want to test.
4. Double-click the service you want to test.

   Digital Event Services or Designer displays the configured service in the service template's Adapter Service Editor.
5. Perform one of the following
   - If you are using Digital Event Services, select **Test > Run**.
   - If you are using Designer, select **Run > Run As > Run Service**.
6. For every service input field, you will be prompted to enter an input value. Enter a value for each input field and then click **OK**.
7. Click the **Results** tab to view the output from this service.

Viewing Adapter Services

You use Digital Event Services or Designer to view adapter services.

To view an adapter service

1. Review the steps in "Before Configuring or Managing Adapter Services" on page 72.
2. In Digital Event Services or Designer, expand the package and folder that contain the service you want to view.
3. Double-click the service you want to view.

   Digital Event Services or Designer displays the configured service in the service template's Adapter Service Editor.

Editing Adapter Services

You use Designer to edit adapter services.

To edit an adapter service

1. In Designer, browse to and open the adapter service that you want to edit.

2. Double-click the service that you want to edit.

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

3. 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**.
   - If you are using the local service development feature, from the **Team** menu in Designer, select the appropriate option to check out the service. The options available in the **Team** menu depend on the VCS client that you use.

4. 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.

5. After you complete 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**.
   - If you are using the local service development feature, from the **Team** menu in Designer, select the appropriate option to check in the service. The options available in the **Team** menu depend on the VCS client that you use.

6. Save the service.

Deleting Services

You use Digital Event Services or Designer to delete adapter services.

To delete a service

1. Review the steps in "Before Configuring or Managing Adapter Services" on page 72.

2. In Digital Event Services 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

You can use Digital Event Services or Designer to validate the values for all adapter services or for a single adapter service.

Validating Values for All Adapter Services

To always validate the values for all adapter services

1. Review the steps in "Before Configuring or Managing Adapter Services" on page 72.
2. Start Digital Event Services or Designer.
3. Perform one of the following:
   - If you are using Digital Event Services, 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**.

The **Automatic data validation** option enables data validation for the selected adapter service only. It compares the service values against the resource data that has already been fetched from the adapter. Note that this option can slow operations.

For more information about the **Adapter Service/Notification Editor**, see the *webMethods Service Development Help* for your release.

Validating Values for a Single Adapter Service

To enable automatic data validation for a single adapter service

1. Review the steps in "Before Configuring or Managing Adapter Services" on page 72.
2. In Digital Event Services 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.

   Digital Event Services or Designer displays the configured adapter service in the service template's Adapter Service Editor.

4. Click the icon.

Reloading Adapter Values

You can enable the Siebel Adapter to reload and validate user-defined data for adapter services at design time in Digital Event Services or Designer. You can reload values for a single adapter service or you can configure Digital Event Services 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, Digital Event Services or Designer compares the service values against the resource data that has already been fetched from the selected adapter.

Reloading Values for All Adapter Services

To reload the adapter values for all adapter services

1. Review the steps in "Before Configuring or Managing Adapter Services" on page 72.

2. Start Digital Event Services or Designer.

3. Perform one of the following:

   - If you are using Digital Event Services, 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**.

Note:
For more information about the **Adapter Service/Notification Editor**, see the *webMethods Service Development Help* for your release.

Reloading Values for a Single Adapter Service

To reload the adapter values for a single adapter service

1. Review the steps in "Before Configuring or Managing Adapter Services" on page 72.

2. In Digital Event Services 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.

   Digital Event Services or Designer displays the configured adapter service in the service template's Adapter Service Editor.

4. Click the icon.

6 Siebel-To-webMethods Communication

Overview

There are three ways to accomplish Siebel-to-webMethods communication:

- EAI webMethods Transports
- Siebel Visual Basic (VB) scripts
- HTTP posting to Integration Server using the Siebel business service named EAI HTTP Transport.

EAI webMethods Transports

As described in "EAI webMethods Transports" on page 31, the EAI webMethods Transports are custom Siebel business service methods.

You can use transports to perform the following tasks:

To ...	See ...
Synchronously invoke a service on Integration Server, using non-guaranteed delivery mode, and return data	"Synchronous IS Service Invoke" on page 99.
Synchronously invoke a service on Integration Server, using guaranteed delivery mode, and return data	"Guaranteed Synchronous IS Service Invoke" on page 101.
Asynchronously invoke a service on Integration Server, using guaranteed delivery mode, and return data	"Guaranteed Asynchronous IS Service Invoke" on page 103.
Get a transaction's status	"Get Transaction Status" on page 105.
Get a transaction's data	"Get Transaction Data" on page 106.
Restart a transaction	"Restart Transaction" on page 107.
End a transaction	"End Transaction" on page 108.
Report exceptions that occur when invoking Integration Server services in asynchronous mode	"webMethods IS Transport Exception" on page 108.

Using the Transport Configuration File

The Siebel Adapter provides a configuration file for the transports, named WmISTransportCfg.txt. This self-documented file is located in the directory in which you installed the transports. Typically, you install transports in *SiebelServer_directory*\siebsrvr. For more information about installing transports, see "Installing, Upgrading, and Uninstalling the Siebel Adapter " on page 37.

Use this file to specify:

- A connection login string that the Guaranteed Asynchronous IS Service Invoke transport uses to establish a Siebel Java Thin client (Java Data Bean) connection to the Siebel Server. This transport will use this connection to return data from Integration Server.
- A directory for storing the guaranteed delivery files for the guaranteed delivery transports.
- Connection pool parameters.
  - For the synchronous service invocation transports (both guaranteed and non-guaranteed), you configure a pool of connections to Integration Server. These transports use this connection pool to invoke Integration Server services on Integration Server.
  - For the Guaranteed Asynchronous IS Service Invoke transport, you configure:
    - A pool of connections to Integration Server. This transport uses this connection pool to invoke Integration Server services on Integration Server.
    - A pool of Java Data Bean connections to the Siebel Server. This transport uses this connection pool to return data from the Integration Server service to the Siebel Server.

Synchronous IS Service Invoke

- **Display name:** Synchronous IS Service Invoke
- **Business service method name:** SyncISInvoke
- **Description:** Synchronously invokes an Integration Server service on Integration Server in a non-guaranteed delivery invocation mode.

Note:
Be sure to specify values for the appropriate parameters in the WmISTransport.cfg file, as described in "Using the Transport Configuration File" on page 98.

Note:
Siebel Adapter provides a sample method that you can use for "Testing Synchronous IS Service Invoke" on page 100, as described on "Testing Synchronous IS Service Invoke" on page 100.

Inputs:

Input Parameter	Display Name	Description
certificates	Certificates	Optional. A semicolon-separated string specifying the absolute path and names of all the certificates in your certificate chain.
delimiter	Delimiter	Optional. A user-specified delimiter character used to separate input Keys from Values. The default is a semi-colon. Use a different delimiter if the data you are passing to webMethods contains a semi-colon.

Input Parameter	Display Name	Description
folderName	Folder	The Integration Server folder that contains the service to invoke. Separate subfolders with a period.
inputData	Input Data	Optional. A property set containing data being sent to Integration Server.
password	Password	The password used to connect to Integration Server.
privateKey	Private Key	Optional. A string specifying the absolute path and name of the private key that corresponds to the public key in your certificate.
server	IS Server Name	A string specifying Integration Server's name in the form `hostname:portNumber`.
serviceName	Service Name	The service you want to invoke on Integration Server.
userName	User Name	The user name used to connect to Integration Server.
useSSL	Use SSL	A string (Yes/No) indicating whether you are using SSL to pass data securely from Siebel to Integration Server. Using SSL requires a corresponding port on Integration Server that can handle an SSL connection to Integration Server.

Outputs:

Output Parameter	Display Name	Description
message	Message	A description of either the success or failure of the operation.
outputData	Output Data	A property set containing data received from Integration Server. The data returned from Integration Server is contained in a child Siebel property set, while the message and status parameters are contained in the parent Siebel property set.
status	Operation Status	A string specifying the status of the operation.

Testing Synchronous IS Service Invoke

The Siebel Adapter provides a sample method that invokes the Synchronous IS Service Invoke transport. It invokes the addInts service provided by Integration Server. The addInts service sums two numbers and displays the result. You can run this sample method in the Siebel Business Service Simulator.

- **Display name:** Test Synchronous IS Service Invoke
- **Business service method name:** TestSyncISInvoke

Inputs:

Input Parameter	Display Name	Description
password	Password	The password used to connect to Integration Server.
server	IS Server Name	A string specifying Integration Server's name in the form hostname:portNumber.
userName	User Name	The user name used to connect to Integration Server.

Note:
The required input parameters folderName and serviceName are hard coded with the values pub.math and addInts, respectively.

Outputs:

Output Parameter	Display Name	Description
message	Message	A description of either the success or failure of the operation.
outputData	Output Data	A property set containing data received from Integration Server. The data returned from Integration Server is contained in a child Siebel property set, while the message and status parameters are contained in the parent Siebel property set.
status	Operation Status	A string specifying the status of the operation.

Guaranteed Synchronous IS Service Invoke

- **Display name:** Guaranteed Synchronous IS Service Invoke
- **Business service method name:** SyncISInvokeG

- **Description:** Synchronously invokes an Integration Server service on Integration Server via the Integration Server guaranteed delivery facility.

Note:
Be sure to specify values for the appropriate parameters in the WmISTransport.cfg file, as described in "Using the Transport Configuration File" on page 98.

Inputs:

Input Parameter	Display Name	Description
certificates	Certificates	Optional. A semi-colon-separated string specifying the absolute path and names of all the certificates in your certificate chain.
delimiter	Delimiter	Optional. A user-specified delimiter character used to separate input Keys from Values. The default is a semi-colon. Use a different delimiter if the data you are passing to webMethods contains a semi-colon.
folderName	Folder	The Integration Server folder that contains the service to invoke. Separate subfolders with a period.
inputData	Input Data	Optional. A property set containing data being sent to Integration Server.
numRetry	Number of Retries	The number of retries allowed to invoke a service on Integration Server.
password	Password	The password used to connect to Integration Server.
privateKey	Private Key	Optional. A string specifying the absolute path and name of the private key that corresponds to the public key in your certificate.
server	IS Server Name	A string specifying Integration Server's name in the form hostname:portNumber.
serviceName	Service Name	The service you want to invoke on Integration Server.
timeToLive	Time to Live	A string specifying this transaction's Time to Live, in minutes.
userName	User Name	The user name used to connect to Integration Server.

Input Parameter	Display Name	Description
useSSL	Use SSL	A string (Yes/No) indicating whether you are using SSL to pass data securely from Siebel to Integration Server. Using SSL requires a corresponding port on Integration Server that can handle an SSL connection to Integration Server.

Outputs:

Output Parameter	Display Name	Description
message	Message	A description of either the success or failure of the operation.
outputData	Output Data	A property set containing data received from Integration Server. The data returned from Integration Server is contained in a child Siebel property set, while the message and status parameters are contained in the parent Siebel property set.
status	Operation Status	A string specifying the status of the operation.

Guaranteed Asynchronous IS Service Invoke

- **Display name:** Guaranteed Asynchronous IS Service Invoke
- **Business service method name:** AsyncISInvokeG
- **Description:** Asynchronously invokes an Integration Server service on Integration Server via the Integration Server guaranteed delivery facility.

Note:
Be sure to specify values for the appropriate parameters in the WmISTransport.cfg file, as described in "Using the Transport Configuration File" on page 98.

Note:
To report exceptions that occur when executing this transport, see "webMethods IS Transport Exception" on page 108.

Inputs:

Input Parameter	Display Name	Description
busServiceDetails	Siebel Business Service	The names of the Siebel business service and method to invoke to pass data received from webMethods to Siebel. Use this format: `busServiceName:methodName`
certificates	Certificates	Optional. A semi-colon-separated string specifying the absolute path and names of all the certificates in your certificate chain.
delimiter	Delimiter	Optional. A user-specified delimiter character used to separate input Keys from Values. The default is a semi-colon. Use a different delimiter if the data you are passing to webMethods contains a semi-colon.
folderName	Folder	The Integration Server folder that contains the service to invoke. Separate subfolders with a period.
inputData	Input Data	Optional. A property set containing data being sent to the service to Integration Server.
numRetry	Number of Retries	The number of retries allowed to invoke a service on Integration Server.
password	Password	The password used to connect to Integration Server.
privateKey	Private Key	Optional. A string specifying the absolute path and name of the private key that corresponds to the public key in your certificate.
sblConnTimeOut	Siebel Connection Timeout	Optional. The number of milliseconds that the transport will wait to establish a Java Thin client connection to the Siebel Server. This connection is used to return data from Integration Server.
sblPassword	Siebel Password	The password used to connect a Java Thin client to the Siebel Server.
sblUserName	Siebel User Name	The user name used to connect a Java Thin client to the Siebel Server.
server	IS Server Name	A string specifying the Integration Server's name in the form hostname:portNumber.

Input Parameter	Display Name	Description
serviceName	Service Name	The service you want to invoke on Integration Server.
timeToLive	Time to Live	A string specifying this transaction's Time to Live, in minutes.
transactionID	Transaction ID	An automatically generated string that identifies the transaction.
userName	User Name	The user name used to connect to Integration Server.
useSSL	Use SSL	A string (Yes/No) indicating whether you are using SSL to pass data securely from Siebel to Integration Server. Using SSL requires a corresponding port on Integration Server that can handle an SSL connection to Integration Server.

Outputs:

Output Parameter	Display Name	Description
message	Message	A description of either the success or failure of the operation.
status	Operation Status	A string specifying the status of the operation. If the transaction fails, the status is returned as 0.

Get Transaction Status

- **Display name:** Get Transaction Status
- **Business service method name:** GetTransactionStatus
- **Description:** Returns the status of a transaction that was started by a Guaranteed Synchronous IS Service Invoke or Guaranteed Asynchronous IS Service Invoke transport.

Inputs:

Input Parameter	Display Name	Description
txID	Transaction ID	Optional. A string that specifies the transaction ID for which you are trying to retrieve the status. This method retrieves the status of all

Input Parameter	Display Name	Description
		transactions when a Transaction ID is not specified.

Outputs:

Output Parameter	Display Name	Description
message	Message	A description of either the success or failure of the operation.
outputData	Output Data	A property set containing data received from Integration Server. The data returned from Integration Server is contained in a child Siebel property set, while the message and status parameters are contained in the parent Siebel property set.
status	Operation Status	A string specifying the status of the operation.

Get Transaction Data

- **Display name:** Get Transaction Data
- **Business service method name:** GetTxData
- **Description:** Returns data that was returned from Integration Server but not passed back to Siebel during execution of a Guaranteed Asynchronous IS Service Invoke transport.

Inputs:

Input Parameter	Display Name	Description
txID	Transaction ID	A string that specifies the transaction ID for which you are trying to retrieve the data.
delimiter	Delimiter	A user-specified delimiter character used to separate input Keys from Values. The default is a semi-colon. Use a different delimiter if the data you are passing to webMethods contains a semi-colon.

Outputs:

Output Parameter	Display Name	Description
message	Message	A description of either the success or failure of the operation.
outputData	Output Data	A property set containing data received from Integration Server. The data returned from Integration Server is contained in a child Siebel property set, while the message and status parameters are contained in the parent Siebel property set.
status	Operation Status	A string specifying the status of the operation. If the transaction fails, the status is returned as 0.

Restart Transaction

- **Display name:** Restart Transaction
- **Business service method name:** RestartTransaction
- **Description:** Restarts a failed Synchronous or Asynchronous IS Service Invoke (guaranteed or non-guaranteed) transport. Do not try to restart a completed transaction; unexpected behavior may result.

Inputs:

Input Parameter	Display Name	Description
txID	Transaction ID	A string that identifies the transaction that you are restarting.

Outputs:

Output Parameter	Display Name	Description
message	Message	A description of either the success or failure of the operation.
status	Operation Status	A string specifying the status of the operation.

End Transaction

- **Display name:** End Transaction
- **Business service method name:** EndTransaction
- **Description:** Ends a transaction.

Inputs:

Input Parameter	Display Name	Description
txID	Transaction ID	A string that identifies the transaction that you are ending.

Outputs:

Input Parameter	Display Name	Description
message	Message	A description of either the success or failure of the operation.
status	Operation Status	A string specifying the status of the operation.

webMethods IS Transport Exception

- **Display name:** webMethods IS Transport Exception
- **Business service method name:** WebmISTransportException
- **Description:** Reports exceptions that occur when using the Guaranteed Asynchronous IS Service Invoke transport. It writes the exceptions to the wmTransport\ErrorLogs directory on your Siebel Server.

Inputs:

None.

Outputs:

Output Parameter	Display Name	Description
message	Message	A description of either the success or failure of the operation.
status	Operation Status	A string specifying the status of the operation.

Example Transport Invocation Function

This example function invokes the SyncISInvoke transport. Each time an Insert service adds a record to the Contact business component, the function sends the record's first and last names to Integration Server.

```
1.  function BusComp_WriteRecord ()
2.  {
3.     // When an Insert service adds a record into Siebel, send the
4.     // contact's First Name and Last Name to Integration Server.
5.
6.     var inpts = TheApplication().NewPropertySet();
7.     var otpts = TheApplication().NewPropertySet();
8.
9.     inpts.SetProperty("server","localhost:5555");
10.    inpts.SetProperty("userName","Administrator");
11.    inpts.SetProperty("password","manage");
12.    inpts.SetProperty("folderName","myFolder");
13.    inpts.SetProperty("useSSL","No");
14.    inpts.SetProperty("delimiter",":");
15.    inpts.SetProperty("serviceName", "receiveContactInfo");
16.
17.    var inputData = TheApplication().NewPropertySet();
18.    inputData.SetProperty("Last Name",this.GetFieldValue("Last Name"));
19.    inputData.SetProperty("First Name",this.GetFieldValue("First Name"));
20.    inpts.AddChild(inputData);
21.
22.    var oBS = TheApplication().GetService("EAI webMethods Transports");
23.    oBS.InvokeMethod("SyncISInvoke", inpts, otpts);
24.
25.    // Retrieve the invocation status, and message.
26.    var st = otpts.GetProperty("status");
27.    var msg = otpts.GetProperty("message");
28.    // Retrieve the property set containing the data returned by the
29.    // Integration Server.
30.    var data = otpts.GetChild(0);
31.
32. }
```

In this example, note that:

- Line 6 creates the input property set that the function will use to send input data to Integration Server.
- Line 7 creates the output property set that the function will use to receive output data from Integration Server.
- Lines 9-15 define the inputs of the input property set.
- Line 17 assigns the input property set as the value of the transport's inputData parameter.
- Lines 18-19 pass the first and last names of the new contact to the property set.
- Line 20 creates the child property set containing data.
- Lines 22-23 invoke the SyncISInvoke transport.

- Lines 26-27 retrieve the status and message variables from the output property set. These variables are contained in the parent Siebel property set.
- Lines 30 retrieves the child property set from the output property set. The child property set contains the data returned from Integration Server.

Siebel Visual Basic Scripts

As described in "Siebel Visual Basic Scripts" on page 32, you can write a Siebel Visual Basic script to call adapter services and pass values to them. For example, you might create a Siebel procedure that notifies an Integration Server client when a new sales contact is created.

The Siebel Adapter provides a Siebel VB script that you can use as a template for writing your VB scripts.

To write a Siebel VB script

1. Determine where you want to place the VB script. See the *Tools Guides* on the Siebel Bookshelf for information about where to attach scripts.

2. Using Siebel Tools, edit the Siebel application object to which you want to attach the Siebel VB script.

3. Copy the SiebelToISTemplate.txt template to the Siebel VB script and modify the template as needed. The template, which includes in-line comments indicating the areas you should modify, is located in:

   Integration Server_directory \packages\WmSiebelAdapter\templates\SiebelToISTemplate.txt

   When writing a Siebel VB script, use the following basic format:

   a. Retrieve information from the Siebel business component record.

   b. Create a webMethods Values object for service input parameters.

   c. Use the information retrieved from the Siebel business component to populate the input parameters Values object.

   d. Establish a connection to Integration Server.

   e. Invoke an adapter service on the connection.

   f. Check the Values object returned by the adapter service.

   g. End the connection to Integration Server.

Performance Tuning Tips For Siebel Systems

Improving the performance of your Siebel system will probably improve the performance of the Siebel Adapter. Following are general tips for improving the performance of your Siebel system:

- Resonate-enable your Siebel Server
- Increase the number of the Oracle db cursors on your system, as recommended by Siebel
- Configure your Siebel Server to use more than one empty server, and ensure that each empty server handles more than the minimum number of tasks

A Logging and Exception Handling

Overview

The following sections describe Siebel Adapter message logging, exception handling, and error codes (including cause and response information).

Adapter Logging Levels

The Siebel Adapter uses the Integration Server logging mechanism to log messages. You can configure and view the Integration Server logs to monitor and troubleshoot the Siebel Adapter. For detailed information about logging into Integration Server, including instructions for configuring and viewing the different kinds of logs supported by the server, see the *webMethods Integration Server Administrator's Guide* for your release.

Configuring Adapter Logging Levels

Beginning with Integration Server 7.1, you can configure different logging levels for the Siebel Adapter.

Accessing the Adapters Logging Information

To access the adapter's logging information

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

   The **Logging Settings** screen appears. 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.

Changing Logging Settings

To change logging settings for the adapter

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

   The Logging Settings screen appears. From the Logging Settings screen, select **Edit Logging Settings**. Select the required **Level of logging** for the Siebel Adapter.

2. 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 *webMethods Audit Logging Guide* for your release.

Siebel Adapter Message Logging

Integration Server maintains several types of logs. However, the Siebel Adapter logs messages only to the audit, error, and server logs. Because the Siebel 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 Integration Server you are running the adapter on, as shown in the following table.

Integration Server	Log	Description
Integration Server 6.5	Audit Log	You can monitor individual adapter services using the audit log as you would audit any service in Integration Server. The audit properties for an adapter service are available in each Siebel Adapter service template on the **Audit** tab.
	Error Log	The Siebel Adapter automatically posts critical-level and error-level log messages to Integration Server's Error log. These log messages will appear as Adapter Runtime messages.
	Server Log	The Siebel 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 Siebel Adapter log messages.
Integration Server 7.1 or higher	Audit Log	You can monitor individual adapter services using the audit log as you would audit any service in Integration Server. The audit properties for an adapter service are available in the Siebel Adapter service template on the **Audit** tab.
	Error Log	The Siebel Adapter automatically posts fatal-level and error-level log messages to the error log. These log messages will appear as Adapter Run-time messages.
	Server Log	Siebel 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 Run-time log messages. Trace-level log messages appear as Siebel Adapter log messages.

Siebel Adapter log messages appear in the following format: ADA.0700.*nnnnc*, where:

- ADA is the facility code that indicates that the message is from an adapter.

- 0700 is the Siebel Adapter major code, which indicates that the error is generated by the Siebel Adapter.
- *nnnn* represents the error's minor code. For detailed descriptions of the Siebel Adapter's minor codes, see " Siebel Adapter Error Codes" on page 116.
- *c* represents the message's severity level (optional).

To monitor Siebel 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)

Siebel Adapter Exception Handling

If a Siebel Adapter object (for example, a connection or service) encounters an error with the Siebel system, it will throw an adapter error coupled with the Siebel error, exactly as it was thrown by Siebel. The Siebel errors will be in an IData format.

For example, if a Query service fails on the Siebel system at run time because its search expression contains invalid values, you will receive an adapter error that indicates that the Query service failed, and the adapter error will contain the specific error generated on the Siebel system indicating why the service failed. In this case, you would receive a Siebel error specifying that the search expression values are invalid.

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 these types of exceptions.

Siebel Adapter Error Codes

The following table lists Siebel Adapter error codes, and provides information about the cause and suggested response for each error.

Error Code	Description
1000	*variable text* **threw a SiebelException: code(***ErrorCode***), msg(***Message***).** *ErrorCode* is the Siebel exception code and *Message* is the exception message returned by Siebel.
	Explanation: A SiebelException occurred.
	Action: Varies according to cause.

Error Code	Description
1001	*variable text* **threw: type(***ExceptionType***), msg(***Message***).** ***ExceptionType* is the type of exception and *Message* is the exception message returned by the Siebel Adapter.**
	Explanation: An AdapterException has occurred. This error code is reserved for all exceptions that do not have a specific exception message listed below.
	Action: Varies according to cause.
1002	**Invalid method inputs.**
	Explanation: Invalid inputs to the method.
	Action: Retry the operation with valid inputs.
1003	**An MVL field must be specified at each level in order to support the next query level.**
	Explanation: An internal error occurred with an adapter service that uses a multi-valued link (MVL) business component.
	Action: Ensure that the adapter service configuration includes an MVL business component in the **Navigation Path** tab.
1004	**Unable to {***variable text***} fields.**
	Explanation: An exception was thrown when the adapter could not deactivate business component fields before executing a Query on the business component.
	Action: Check the business component definition to ensure that the specified fields are included in the business component, and that there are no restrictions on the fields that may prevent the adapter from deactivating the field.
1005	**Unable to extract fields from the business component after a Query operation.**
	Explanation: The operation failed, throwing a SiebelException: code(*ErrorCode*), msg(*Message*). Indicates a failure in retrieving business component fields after executing a Query.
	Action: Ensure that the specified fields are included in the business component definition, and that there are no restrictions on the fields that may prevent the adapter from reading the field values.
1006	**Cannot set Query on the business component when performing an Insert operation.**
	Explanation: Missing search expressions for the selected multi-value link and additional business components.

Error Code	Description
	Action: Using the **Search** tab, define search expressions for the selected multi-valued link and additional business components.
1007	**Select {*variable text*} on the Navigation Path tab before you specify search criteria for it.**
	Explanation: You did not select {*variable text*} on the **Navigation Path** tab.
	Action: Select {*variable text*} on the **Navigation Path** tab before you specify search criteria for it on the **Search** tab.
1008	**Threw a non-fatal exception: type(*ErrorType*), msg(*Message*).** ***ErrorType* is the type of exception and *Message* is the exception message returned by the Siebel Adapter.**
	Explanation: A non-fatal AdapterException has occurred.
	Action: No response is necessary. For example, setRepositoryContext() throws an exception since it cannot be called on certain business components. This exception is logged, and does not require any user action.
1009	**The placeholder {*variable text*} does not correspond to a field in the input**
	Explanation: The name of an input variable does not match the placeholder in the query specification.
	Action: Save the change you made to the input variable name.
1010	**No records were returned as a result of the Query; unable to perform {*variable text*}.**
	Explanation: Query failed to return one or more records when performing one of the following operations: Delete, Update, or Associate.
	Action: Check the adapter service input and the search criteria specified in the **Search** tab.
1011	**Note:** Message 1011 is intentionally omitted.
1012	**Must select two business components for an Associate operation.**
	Explanation: You failed to select two business components.
	Action: Select two business components in the **Navigation Path** tab, and specify a search expression for each one in the **Search** tab.
1013	**Business service operation input, {*variable text*}, is a required value.**

Error Code	Description
	Explanation: The names of a Business Service and a Business Service method are required parameters when executing the BusinessServiceOperation adapter service template.
	Action: Specify the names of a Business Service and a Business Service method in the **Business Service Inputs** field on the **Siebel Business Service Inputs/Outputs** tab.
1014	**The number of input keys and values must match.**
	Explanation: The arrays of Siebel property set keys and Siebel property set values that you selected contain a different number of keys and values.
	Action: In the **Business Service Inputs** field on the **Siebel Business Service Inputs/Outputs** tab, ensure that the array of Siebel property set keys and the array of Siebel property set values contain the same number of keys and values.
1015	**There was no response from the Siebel Server.**
	Explanation: An existing adapter connection object could not communicate with the Siebel Server, perhaps due to loss of network connectivity or Siebel Server shutdown. **Action: The adapter will attempt to clean up the Siebel Adapter connection pool and re-connect to the Siebel Server.**
	Action: If the problem persists, restart the Siebel Adapter.
1016	**Invalid connection; please retry operation.**
	Explanation: The adapter received a null connection object.
	Action: Retry the operation. If the problem persists, restart your adapter connection using Integration Server Administrator.
1017	**Threw a SiebelException: code ({*ErrorCode*}), msg ({*Message*}). The connection pool will now be recycled. Please retry your operation.**
	Explanation: An existing adapter connection object could not communicate with the Siebel Server, perhaps due to loss of network connectivity or Siebel Server shutdown. The BOI API returns an error code and error message indicating this fact.
	Action: Retry the operation. If the problem persists, restart the Siebel Adapter.
1018	**Unable to get {*variable text*}. Not connected to Siebel Server; please check your network connection and make sure the Siebel Server is running.**
	Explanation: Occurs when a stale connection is retrieved from the connection pool.

Error Code	Description
	A connection becomes stale when the adapter loses a physical connection to the Siebel Server, but the adapter considers the connection to be active. This can occur when the network connection has gone down or the Siebel Server shuts down unexpectedly.
	Action: The adapter removes the stale connection and recycles that connection's pool. That is, it shuts down the connection pool and re-starts it. If this problem persists, please reset your Siebel Server connections.
1019	**One or more inputs are missing, Please check your inputs and retry the operation.**
	Explanation: Improper inputs were passed to the AttachmentOperation or the InvokeBusinessComponent adapter service template.
	Action: For an Attachment adapter service, check the values on the **Attachment Operation Inputs** tab. For an Invoke Business Component adapter service, check the values on the **Invoke Business Component Method - Inputs** tab.
1020	**Siebel Attachment operation returned: {variable text}.**
	Explanation: Indicates whether the Attachment operation succeeded.
	Action: None required.

B Adapter Configuration Parameters Appendix

Overview

This appendix contains a description of Siebel Adapter parameters, you can specify in the server configuration file (server.cnf), which is located in the *Integration Server_directory* `\config` directory. Typically you use the **Settings > Extended Settings** screen in Integration Server Administrator to update this file, but there might be times when you need to edit the file directly using a text editor. If you edit the file directly, you should first shut down Integration Server before updating the file. After you make the changes, restart the server. If you are using the **Settings > Extended Settings** screen to update the server configuration file (server.cnf), a server restart is not required unless otherwise specified. The server uses default values for the parameters. If a parameter has a default, it is listed with the description of the parameter.