Siebel Connector for Siebel *e*Business Applications

Version 7.0, Rev. H

12-BCK98F

August 2002

Contents

Chapter 3. Using the Connector

Chapter 4. Customizing the Connector

Chapter 5. Initializing a New Instance

Chapter 6. Administration and Error Messages

Index

Introduction

This guide provides information on using *Siebel Connector for Siebel eBusiness Applications*. This guide will be useful primarily to people whose title or job description matches one of the following:

Business Analysts	Persons responsible for analyzing application integration challenges and planning integration solutions at an enterprise.
Siebel Application Administrators	Persons responsible for planning, setting up, and maintaining Siebel applications.
Siebel Application Developers	Persons responsible for planning, implementing, and configuring Siebel applications, and possibly adding new functionality.
Siebel Integration Developers	Persons responsible for analyzing a business situation or using the analysis of a Business Analyst to build the integration solution at an enterprise for Siebel Applications.
Siebel System Administrators	Persons responsible for the whole system, including installing, maintaining, and upgrading Siebel products.
System Integrators	Persons responsible for analyzing a business situation or using the analysis of a Business Analyst to build the integration solution at an enterprise for specific applications and or to develop custom solutions.

The audience for this book also includes any user with experience in data integration, data transformation (data mapping), scripting or programming, and XML.

How This Guide Is Organized

This guide provides information necessary to implement and administer Siebel Connector. The information is organized into the following types of chapters:

- Reference—provides an overview of Siebel Connector and features.
- Procedures—provides procedures to implement and use the features of Siebel Connector.
- Administration—provides procedures to administer Siebel Connector.

Revision History

Siebel Connector for Siebel eBusiness Applications, Version 7.0, Rev. H.

Overview 1

Siebel Connector for Siebel eBusiness Applications provides a prebuilt solution for replicating global data between two or more separate Siebel instances (Siebel 7 and above). The connector uses Siebel eAI and IBM MQSeries or Microsoft Message Queuing to connect these instances together.

Many organizations have several separate Siebel instances, distributed either geographically or across multiple business units. For example, your company's North American organization may have implemented Siebel Sales and Field Service, while your European organization may have implemented Siebel Call Center and eChannel. However, for your global enterprise, it is critical to track and maintain customers and their contacts and activities across all these instances to provide a unified view of customer information irrespective of the geographical boundaries. The *Siebel Connector for Siebel eBusiness Applications* is designed to help keep global data consistent across all of these instances for a unified view of customer information. Figure 1 shows a deployment among three separate Siebel instances.

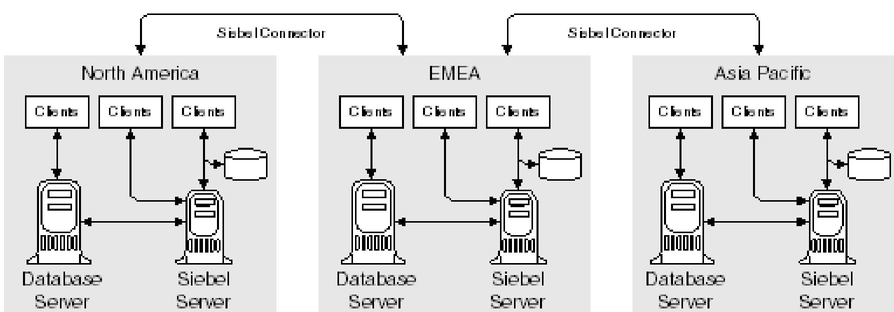


Figure 1. Siebel Connector Deployment Among Three Instances

Using the Connector

You can approach your work with the *Siebel Connector for Siebel eBusiness Applications* in the following ways, depending upon your specific needs:

- Use the Siebel standard integrations for global accounts as is.
- Modify the standard integration to suit your business needs.
- Create customized integrations to support your own business needs.

You can learn how to use the standard integrations, as is, from this guide. You can also find guidance on modifying these standard integrations. More information on creating customized integrations is included, but you also need to consult Siebel eBusiness Application Integration books for more details on Siebel eAI.

The *Siebel Connector for Siebel eBusiness Applications* provides:

- A connector to replicate global accounts between multiple Siebel 7 and above instances.
- XML-based publishing and subscribing to configure and synchronize any Siebel object.
- Preconfigured integration object for Accounts, including Account, Contacts, Activities, and Business Address information.
- Support for data transformation using Siebel Data Mapper
- Support for third-party middleware such as IBM MQSeries and MSMQ.

Concepts and Terminology

The *Siebel Connector for Siebel eBusiness Applications* introduces several new components. It is important to understand the concepts and the terms and functionality of each of these components before embarking on a detailed design.

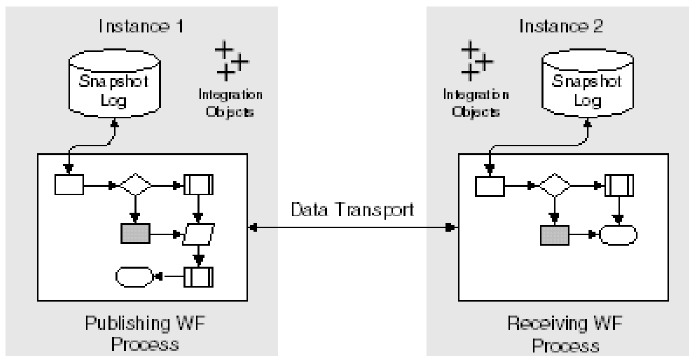


Figure 2. Connection Between Two Siebel Instances

The *Siebel Connector for Siebel eBusiness Applications* provides selected replication between multiple separate Siebel instances using a master and slave model.

Instances

A given enterprise could have multiple Siebel installations, each being unique versions running on its own database and application servers. In order for global data to be shared among these instances, the instances need to have a unique identifier.

Each instance is uniquely identified using an instance ID. The instance IDs are defined using a screen for defining instances from the *Siebel Connector for Siebel eBusiness Applications* administration screens.

The instance IDs must be:

- Unique within a network of Siebel instances
- Consists of two segments:
  - Major segment—CSN or DUNS number
  - Minor segment—Customer specific
- Set by the customer using the Instance Identification view

  Example of an Instance ID: 808856512-Siebel US

Master and Slave

In a master and slave model one instance is designated as an owner (master) for a particular data record and every other instance that gets a copy of the record is a slave. The master or owner instance can read, update or delete that particular data record. The slave instances have a read only copy of that record.

The *Siebel Connector for Siebel eBusiness Applications* provides selective replication of data. Only global data records are picked up for replication. The publisher on the master instance sends the record to a channel. The subscribers receive the data on this channel.

Global Data

Typically, only a small part of the data needs to be synchronized across the connector network. For example, Siebel instances in different countries may be interested only in sharing the global accounts and not the national or regional accounts. In the connector framework, data that needs to be synchronized across the connector network is marked as Global Data. Each record marked as Global Data should be assigned one Siebel instance as the owner (the master) instance. Changes to global data are allowed only at the master instance and are periodically propagated to all interested instances on the connector network.

Global data:

- Identifies data that needs to be replicated.
- Can be set through the Siebel user interface for Accounts, Contacts, Activities, and Business Addresses.
- A global record is owned by one and only one instance.

Connected users at the owning instance can update, delete, and change the ownership of global data. Users at subscribing instances have read only access to the record.

Publishers and Subscribers

Publishers are workflows in Siebel applications that are scheduled to run periodically. The publisher has a sequence of steps as part of its workflows to identify all global records that need to be replicated for a given channel, compare the record to a previous snapshot to determine the changes made since the last publish and then finally send the information over a transport such as IBM MQSeries or MSMQ. The publisher may include steps to transform the data from one format to another or to process additional changes before sending the data over.

Subscribers are also workflows but these workflows “receive” data that has been published on a particular channel.

Channels

Channels describe the logical path between any two instances. A given channel is common between two instances and hence must have the same name when defined on these two instances. A channel may map to a physical transport destination (a queue in the case of IBM MQ or MSMQ). Channels also provide the collection of objects that any two instances are interested in.

Channel Objects and Integration Objects

A channel is used to communicate one or more conversation topics; a conversation topic in a publish-subscribe paradigm maps to a Siebel integration object. Integration objects are Siebel metadata defining what Siebel data is replicated.

Integration objects also provide an XML representation of the defined subset of Siebel data. For more details on integration objects, see Chapter 4, "Customizing the Connector."

The *Siebel Connector for Siebel eBusiness Applications* includes a predefined integration object for Global Accounts (as detailed in Table 1), with Contacts, Activities, and Business Address as its children. These integration objects can be customized or new ones created using the integration object wizard to meet the business requirements around global data synchronization.

Table 1. Predefined Integration Object for Global Accounts

Integration Component	Business Component	Parent Integration Component
Account	Account	-
Account_Business Address	Business Address	Account
Action	Action	Account
Contact	Contact	Account

Channel Objects and Query Specifications

Query Specifications on the channel objects can be used to subset the data published between multiple instances. Query Specifications can also be used to limit the global data exchange between two instances; for example, a Siebel instance in France may be only interested in global accounts that are either located in Europe or are located in US and are high opportunity accounts.

Channel Objects and Data Map

The structure of the global data in two Siebel instances on the connector network may be different, either because of different business rules in different divisions or different versions. Such differences require that global data needs to be mapped from one structure to another for synchronization across the network. A data map, either a Siebel Data Mapper data map or an eScript data map, can be associated with each channel object. If a data map is specified, it is automatically executed before publishing or after receiving data on that channel. For more details about data mapping, see *Business Processes and Rules: Siebel eBusiness Application Integration Volume IV*.

Incremental Snapshots

Each channel stores a snapshot of data published or received on that channel. These snapshots are used to ensure that only incremental data is published or received, increasing performance and saving bandwidth.

The technology that allows *Siebel Connector for Siebel eBusiness Applications* to work is the Incremental Snapshot. This is a business service that calculates the difference between the current version of an object and a previous version. This allows *Siebel Connector for Siebel eBusiness Applications* to maintain control and integrity of the synchronization process. Incremental Snapshot captures and publishes a complete, or partial image, of the current version of an object so it can be sent to another instance. Incremental snapshot stores the image as an XML document in the snapshot log.

Transport Mechanisms

A channel is a logical connection that ultimately uses a specified transport to implement the communication. The transport mechanism is responsible for actually moving data between the two instances. The transport mechanism is independent of the publish or subscribe logic. The transport mechanism is responsible for providing the guaranteed delivery or other delivery features. Siebel connector is preconfigured to use IBM MQ Series or Microsoft MSMQ as the underlying transport and queuing mechanism.

Comparison with Siebel Replication Manager

Table 2 describes the main differences between Siebel Replication Manager and *Siebel Connector for Siebel eBusiness Applications*.

Table 2. Comparison of Replication Manager and Siebel Connector

Replication Manager	Siebel Connector
Replication occurs at the database or physical layer.	Replication occurs at the application layer, involving the object or component level.
Replication occurs between HQ and regional nodes that must be running in a tightly-coupled architecture at the same time within the same deployment.	Replication occurs between independent deployments of Siebel with their own logical data source. These may or may not be running at the same time. *Siebel Connector for Siebel eBusiness Applications* accommodates different schemas.
Replication Manager is better suited for higher volumes and more difficult routing problems. Requires clearly defined ownership.	*Siebel Connector for Siebel eBusiness Applications* has a limited data set, as the typical volume is approximately 10,000 global accounts.
Middleware is not required.	Middleware is required to transport or publish the message.

Key Design Considerations

The *Siebel Connector for Siebel eBusiness Applications* is designed to address a specific business problem. Therefore, it is important to understand where the connector and its components can be used and where it cannot be used.

The *Siebel Connector for Siebel eBusiness Applications* should be used where:

- The primary business requirement is to replicate selected records for a global object like accounts across multiple Siebel instances.
- Replication is limited to number of global account data across multiple Siebel instances on Siebel 7 and above versions.
- The "master and slave" model of replication meets your business requirements.

The model that the *Siebel Connector for Siebel eBusiness Applications* uses is designed for low volume replication. It is important to understand that this solution is not a substitute for full data replication from one instance to another.

The connector can be customized to replicate data from any Siebel business object as well as use any transport mechanism beneath it.

Setup Topology

When you have more than two instances involved in a *Siebel Connector for Siebel eBusiness Applications* setup, do not use a circular setup and do not create channels between each pair of instances. The possible setups for three instances, 1, 2, and 3, are linear setup and circular setup, as shown in Figure 3.

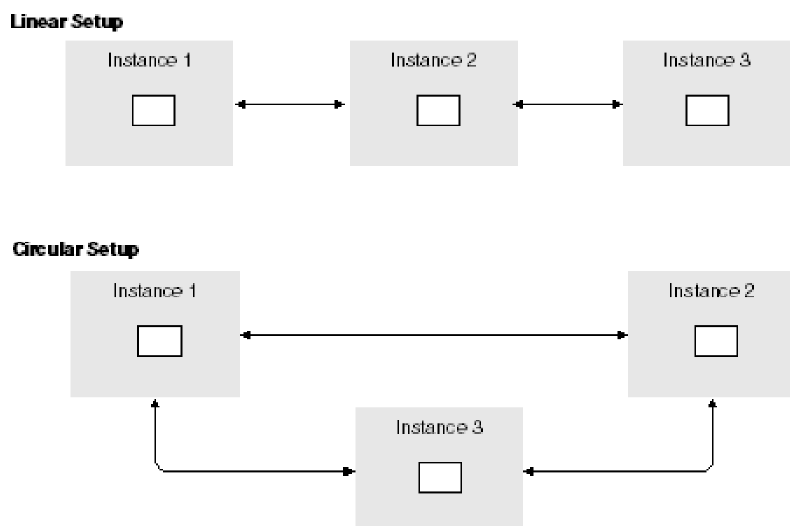


Figure 3. Linear and Circular Setup

CAUTION: Circular setups cause data conflict and data loss because one instance receives the same data from several different channels. There is no synchronization mechanism between channels, therefore older versions of the data can overwrite newer versions. Linear setup eliminates this problem because a given instance receives a particular record from only one source.

Performance

Siebel Connector for Siebel eBusiness Applications performance is determined by various factors—some that Siebel can control but many that depend on hardware, network and other configuration. However, there are optimizations that one can do on the Siebel side to ensure optimal performance. Customizing the integration object provided out of the box for global accounts (Siebel to Siebel: Account) would provide further optimization.

The performance of the *Siebel Connector for Siebel eBusiness Applications* is dependent on the number of components and fields in an integration object. Thus it is important to look at this integration object and determine if all the fields included are really needed for replication in your environment. In most cases you would only need a subset of fields that are included as part of the connector. In these cases you should inactivate those fields in the integration object that you do not want to replicate.

Reducing the size of the integration object greatly increases performance.

Installing and Configuring the Connector 2

This chapter describes the implementation of the *Siebel Connector for Siebel eBusiness Applications* when two or more instances of Siebel eBusiness Applications already exists within the organization, but are not sharing global data. The high-level procedures that you need to complete are listed below.

Checklist

- ❑ Create a publication group.

  For details, see "Creating a Publication Group" on page 24.

- ❑ Define the instance IDs, including the HQ instance ID.

  For details, see "Defining Instance IDs" on page 25 and "Defining the HQ Instance ID" on page 27.

- ❑ Create a channel and associate integration IDs with channels.

  For details, see "Creating a Channel" on page 28 and "Associating Integration Objects with Channels" on page 29.

Creating a Publication Group

This section describes the procedure to create a publication group.

NOTE: Publication groups are not used in Siebel 7, but will be implemented in future releases of Siebel applications for the logical grouping of channels and instances. However, because the publication group is a required field when setting up channel definitions, you still need to set up “dummy” publication groups in Siebel 7.

To create a publication group

1. Navigate to the Siebel to Siebel Connector screen.
2. From the Show drop-down list, choose Publication Groups.

   The Publications Group list appears.
3. Create a new record.
4. Fill in the fields as appropriate.
5. Click Save.
6. Repeat the procedure (starting at Step 3 above) for any remaining publication groups you need to create.

Defining Instance IDs

This procedure describes how to define instance IDs for the instances that will share data with this instance.

NOTE: After defining all the instance IDs, the HQ instance ID will be picked from the instance IDs you have defined. For more information, see "To define the HQ Instance ID for Siebel" on page 27.

To define instance IDs

1. Obtain instance IDs from administrators of other Siebel instances in the deployment.

   NOTE: When selecting an appropriate Major Segment, Siebel recommends using DUNS number. For the Minor Segment, use something descriptive or informative.

2. Navigate to the Siebel to Siebel Connector screen.
3. From the Show drop-down list, select Instances.

   The Instances list appears.

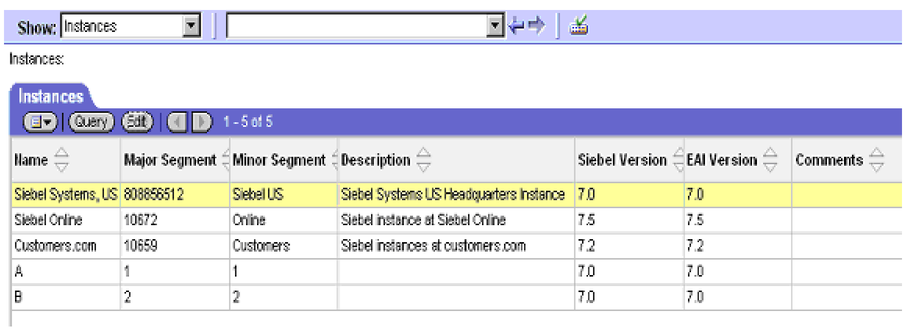
4. Create a new record.

   Fill out the form using the following table as a guide.

Field	Description
Name	Unique name for the instance that has meaning for the SysAdmins.
Major ID segment	Siebel Systems recommends the DUNS #.
Minor ID segment	Anything that is meaningful and is unique among all Siebel Servers that share the same major segment.
Description	A brief description of the Siebel instance.

Field	Description
Siebel Version	The version of Siebel eBusiness Applications running on this server, such as 7.0.
EAI Version	The version of Siebel eBusiness Applications Integration (Siebel eAI) running on this server.
Comments	Any additional pertinent comments, such as configuration information.

5 Click Save to display the Instances list:



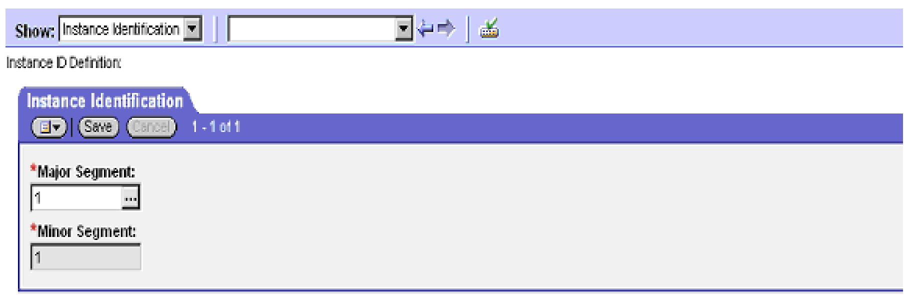
6 Repeat the procedure (starting at Step 4 on page 25) for any remaining instances within the network that will be sharing data with this instance. This includes SysAdmin’s own instance of Siebel.

Defining the HQ Instance ID

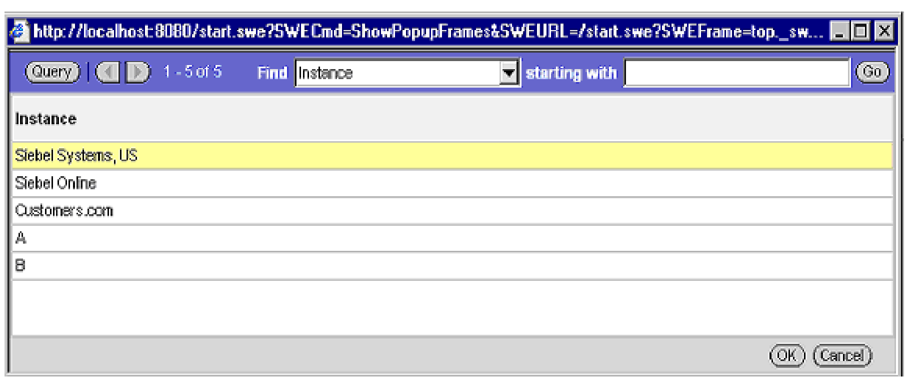
This section describes how to define the HQ Instance ID.

To define the HQ Instance ID for Siebel

1. Navigate to the Siebel to Siebel Connector screen.
2. From the Show drop-down list, click Instances Identification to display the Instances Identification form.



3. Click the ellipsis (...) button and the Pick Instance dialog box appears.



4. Select the instance to be designated the HQ Instance and click OK.

Creating a Channel

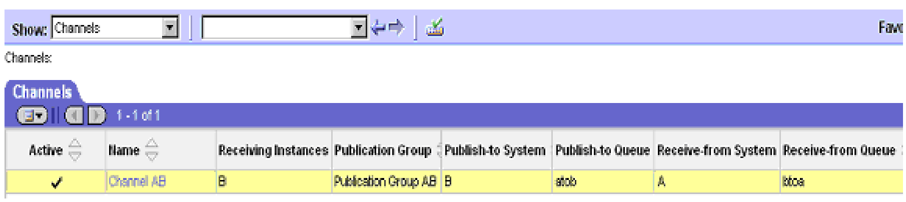
This section describes the procedure to create a channel to link instances. A channel is a logical two-way communication path between two Siebel installations for publishing and receiving data. Two instances sharing global data must have the same channel defined with the same name.

To create a channel

1. Navigate to the Channels view.

   The Channels List applet appears.

2. Create a new record.



3. Fill out the form as appropriate.

   If there is a channel defined between two instances A and B, the Publish-to Queue field for instance A should be the same as the Receive-from Queue field for instance B. Likewise, the Publish-to System for instance A should be the same as Receive-from System on instance B.

   - If you are using the MQSeries Server Transport, the name of the Queue Manager residing on the same machine as siebsrvr is Publish-to System. Publish-to Queue is the name of the Remote queue defined in that Queue Manager.
   - If you are using MSMQ Transport, the machine name where the target queue is located is Publish-to System. Publish-to Queue is the name of the target queue.

4. Click Save.
5. Repeat Step 2 through Step 7 to add as many channels as you need.

Associating Integration Objects with Channels

If necessary, associate pre-existing *Siebel Connector for Siebel eBusiness Applications* Integration Objects with channels to enable the integration objects.

NOTE: For more information about creating additional Integration Objects, see Chapter 4, “Customizing the Connector.” You will need to know what kind of data each channel will replicate.

This section describes the procedure to associate pre-existing Siebel Connector Integration Objects with channels.

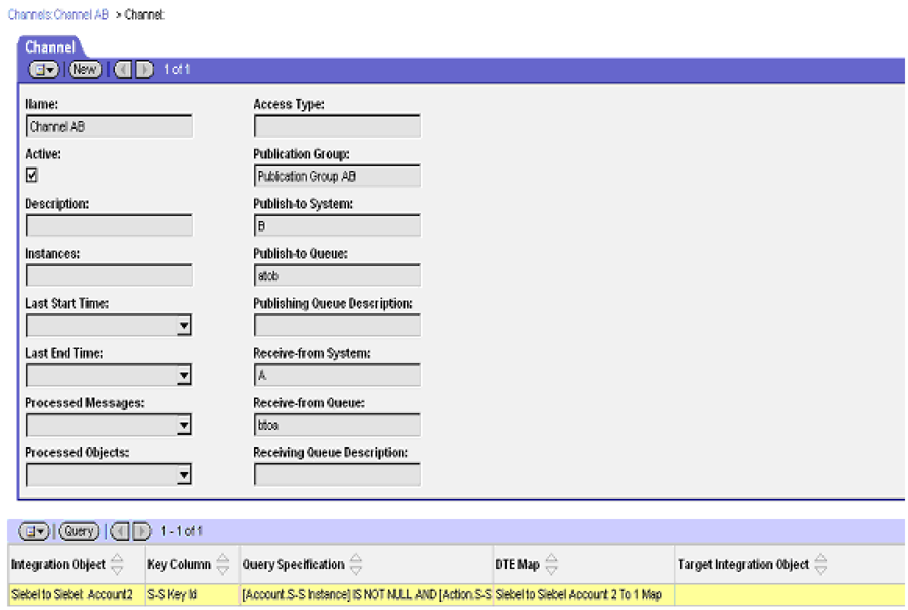
To associate an existing integration object with a channel

1. Navigate to the Channels view.

   The Channels applet appears.

2 In the Name field, click on the name of the channel for the association.

The Channel screen appears:



3 Create a new record on the lower applet.

4 The Channel applet appears with an integration object form below it. Fill out the channel object list as appropriate.

For the Query Specification field, the examples below include a generic spec and then a specific one for an Account.

Generic Query Specification:

```
[<Component_1>.<S-S Instance Field Name>] IS NOT NULL AND
[<Component_2>.<S-S Instance Field Name>] IS NOT NULL AND ...
AND [<Component_n>.<S-S Instance Field Name>] IS NOT NULL
```

For example,

```
[Account.S-S Instance] IS NOT NULL AND [Action.S-S Instance]
IS NOT NULL AND [Account_Business Address.Business Addr S-S
Instance] IS NOT NULL AND [Contact.S-S Instance] IS NOT NULL
```

means “Find all global accounts. For each global account, find its global activities, global business addresses, and global contacts.”

5. For the field DTE Map, select the DTE Map that converts the source integration object to the target integration object.
6. Then, for the Target Integration Object field, select the integration object to be published.
7. Click Save.
8. Repeat this procedure to associate other integration objects with channels.

NOTE: Integration components cannot be shared across channel objects within the same channel. For example, having one integration object containing the contact integration component and a second integration object also containing the contact integration component under a given channel is not allowed.

Using the Connector 3

Data Maps

If the instances in the deployment include different versions of Siebel eBusiness Applications, you must use the Siebel Data Mapper to enable the sharing of global data. However, even with the same version you still need to use the Data Mapper.

NOTE: For details about data transformation, see *Business Processes and Rules: Siebel eBusiness Application Integration Volume IV*.

Transports

To set up and activate a transport adapter, refer to the specific documentation for the vendor you use for this functionality in your network.

Routing Messages

Once a message has been published on a transport, the middleware routes the message to the appropriate instances based on rules defined in the middleware. See the specific documentation from your middleware vendor to set up routing for messages in your network.

Global Data

Global data is important information an organization wants to share through replication, at the application level, between two or more instances of Siebel. Typically, it is approximately 1% to 5% of an organization's entire database. Global data includes information in the form of records from a business object that has a corresponding integration object defined for *Siebel Connector for Siebel eBusiness Applications*. Global data must be explicitly defined because the default for data records is local. The instance owner in the Owner Instance field indicates a record is global and is set through the user interface (UI). Global data can be set based on assignment manager rules when using EIM and other processes and can be shared with other subscribing Siebel instances.

Hierarchy independence of global data is applicable to all objects regardless of their place in the hierarchy. Therefore it cannot be assumed that related components of a global account, such as contacts, opportunities and so on, will also be global.

Subordinate nodes, such as regional or mobile nodes, cannot create a global record. Ownership can be changed by the owner through the UI by adjusting the Instance Name field. An Account could have subordinate objects that may be global or not global. Only global data will be shared. Whenever a global account is created, the record is locked for deletes and updates in non-ownership instances.

Ownership of Global Data

Global data ownership can be designated when the record is created. Any record owned by an instance can be updated or deleted by that instance. Any record not owned by that instance can only be read by that instance.

Transfer of Ownership

Ownership of a global record can be transferred to another instance by the owner. For example, an account is created on an instance US_10001 and after three weeks, the ownership needs to be transferred to Europe_20002. So the account is transferred and it becomes read-only on US_10001 and read/write on Europe_20002. This transfer of ownership can be done through the UI using the Instance Name field.

Changing Global Data

The following describes what changes can and cannot be applied to global data.

Allowable Actions on or with Global Data

Update is only supported on the HQ server in a Replication Manager environment. These changes can transfer ownership to another instance. Insert is supported on all nodes including HQ, regional, and mobile. Global data can only be deleted by the owner.

Actions not Allowed on or with Global Data

Global data cannot be changed to local data. Global data cannot be updated by users on mobile or read-only nodes. Global data cannot be merged with local data or other global data.

Publishing Messages

Publishing is a repetitive server process that periodically runs to publish changes to global data for all subscribing instances. Standard workflows include S2S Outbound Process MQ designed for publishing on IBM MQSeries transports and S2S Outbound Process MSMQ designed for publishing on MSMQ transports. This section describes how to run the outbound workflows.

To publish global records to all channels from the srvrmgr command line

- For IBM MQSeries Server Transport:

  ```
  start task for comp BusIntBatchMgr with ProcessName='S2S Outbound
  Process MQ'
  ```

- For MSMQ Transport:

  ```
  start task for comp BusIntBatchMgr with ProcessName='S2S Outbound
  Process MSMQ'
  ```

NOTE: To schedule publishing using repeating component request, refer to *Siebel Server Administration Guide* for information on how to submit a repeating component request.

Receiving Messages

Receiving is a server component that runs as a background process. It allows the receiving instance to receive messages from publishing instances. You use a server component based on your network's transport mechanism. The connector includes two standard server components to receive inbound messages and dispatch them to the workflow S2S Inbound Process.

The server components are Siebel to Siebel MQSeries Receiver (for IBM MQSeries Server transport) and Siebel to Siebel MSMQ Receiver (for MSMQ transport).

To receive global records from the srvrmgr command line

- For IBM MQSeries Server transport, enter the following commands:

  ```
  create named subsystem <UniqueName> for subsystem
  MQSeriesServerSubsys with MqQueueManagerName=<QueueManagerName>,
  MqPhysicalQueueName=<QueueName>

  start task for comp S2SMqRcvr with
  ReceiverConnectionSubsystem=<UniqueName>
  ```

NOTE: Siebel 7 does not support the IBM MQSeries AMI Receiver.

- For MSMQ Transport, enter the following commands:

  ```
  create named subsystem <UniqueName> for subsystem MSMQSubsys with
  MsmqPhysicalQueueName=<QueueName>,MsmqQueueMachineName=
  <MachineName>

  start task for comp S2SMSMQRcvr with
  ReceiverConnectionSubsystem=<UniqueName>
  ```

Customizing the Connector 4

This chapter describes how to create and modify of integration objects specifically for the Siebel Connector for Siebel eBusiness Applications. This chapter assumes the reader has a general understanding of Enterprise Application Integration (EAI) technology.

Checklist

❑ Create new integration objects.

For details, see "Siebel Connector Integration Objects" on page 43.

❑ Inactivate keys.

For details, see "Deactivating Explicitly Joined Foreign Key Fields" on page 45.

❑ Add ModId fields.

For details, see "Adding ModId Fields" on page 45.

❑ Inactivate primary fields.

For details, see "Deactivating Primary Fields" on page 47.

❑ Add operation fields.

For details, see "Set Type to System." on page 50.

❑ Add status keys.

For details, see "Adding a Status Key" on page 51.

❑ Add field dependencies.

For details, see "Adding Field Dependency" on page 52.

- ❑ Modify data maps, as needed.

  For details, see "Customizing a DTE Map for Use with Siebel Connector" on page 53.

Siebel Connector Integration Objects

This section describes how to create a new integration object for Siebel Connector for Siebel eBusiness Applications.

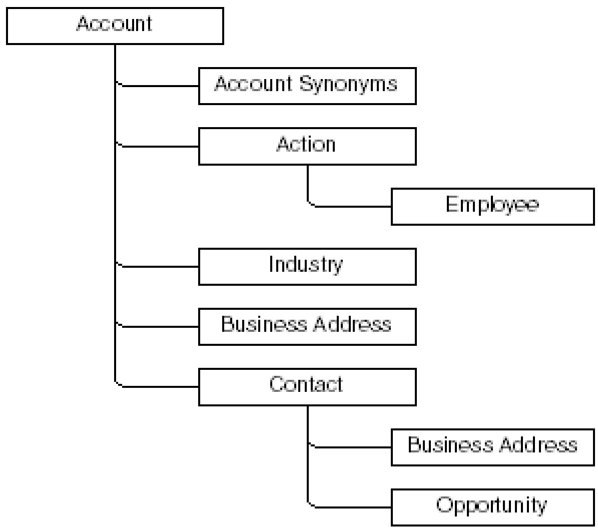


Figure 4. Integration Object Example

NOTE: For more information, see *Integration Platform Technologies: Siebel eBusiness Application Integration Volume II*.

To create new Siebel Connector for Siebel eBusiness Applications integration objects

1. Start Siebel Tools and access the Integration Object Builder wizard.
2. When the Integration Object Builder dialog box appears from within the wizard, choose Siebel to Siebel Connector for the project in which your new integration object will exist.

3. In the same dialog box, choose EAI Siebel Wizard as the business service that interfaces with the source system, and click Next. The second page in the wizard appears.
4. Choose the object that will be the model for the new Siebel integration object.
5. In the same dialog box, enter a unique name in the field for the new integration object.
6. Continue to follow the instructions in the wizard.
7. Complete the instructions and click Finish.

   The new integration object appears in the list of integration objects in Siebel Tools. Repeat this procedure for each new Siebel Connector for Siebel eBusiness Applications Integration Object you require for your deployment.

Integration Object Modification

You must modify new Siebel Connector for Siebel eBusiness Applications Integration Objects. This section describes how to adjust the integration components of the object.

Deactivating Explicitly Joined Foreign Key Fields

You must deactivate some explicitly joined foreign key fields in the integration components to protect data integrity. Do this when the instances in the deployment have different ROW_ID assignments, but the other fields are the same. To properly share global data in this type of Siebel Connector for Siebel eBusiness Applications deployment, you do *not* want the ROW_IDs.

To deactivate explicitly joined foreign key fields

1. Find the corresponding buscomp for the integration component in Siebel Tools.
2. Display the list of joins.
3. Find the foreign key field in the buscomp for each of the joins.
4. Inactivate the corresponding field from the integration component.

Repeat Step 2 through Step 4 for each of the integration components defined in your integration object.

Adding ModId Fields

To improve performance, Siebel EAI Adapter and Incremental Snapshot use ModId fields to determine whether a specific record in a table has been modified. You must add ModId fields to Siebel Connector for Siebel eBusiness Applications integration objects to support Incremental Snapshot. To do this, modify the definition of integration components.

To add ModId fields to integration components

1. In the “table” property of a buscomp definition, find the base table of the corresponding buscomp.

2. Find the extension tables of this base table and identify the tables that require one additional field based on the criterion below.

   Criterion for the addition of one additional field: If the name of the extension table appears in the join property of any of the field definitions for the buscomp, but not in the list when you click on the Join icon in the Object Explorer.

3. For each of the extension tables identified in Step 2, add one field to the integration component definition with a name of <Extension table name>.Mod Id.

   NOTE: This is complex. The main reason to do this is to keep track of the modification numbers of each of base table and extension table for that buscomp. However, you do not need the modification number of the extension table if that table is explicitly joined. Any changes to explicitly joined tables will be recovered by Siebel Adapter when an upsert is called, as long as you keep the definition of the joined field in the buscomp.

   ---

4. For each of the ModId fields (including ModId which is generated automatically by Integration Object Builder) in the integration component definition specify a user property called ModId. Set the value field to Y.

5. Repeat Step 2 on page 46 through Step 4 on page 46 for each integration component defined in the integration object.

   NOTE: If the integration component represents a Multi-Value Group (MVG), add a user property MVGFieldName or AssocFieldName, depending on whether the MVG is associated or not, and set the value to ModId.

   NOTE: To find out whether an MVG has an association, go to the corresponding BusComp/Multi Value Link. Then go to the link definition by drilling down from the Destination Link field. If there is an Inter Table field specified, it is an association and the user property should be set to AssocFieldName = ModId. Otherwise, set the user property to MVGFieldName = Mod Id.

Deactivating Primary Fields

Multi-value Group (MVG) Primary fields should be deactivated in the definition of Siebel Connector for Siebel eBusiness Applications integration objects. The reason is that Siebel Adapter will reconstruct Primary fields if their contents come from the MVG, and the corresponding multi-value link has a non-empty Primary Id Field. Including Primary fields denormalizes the definition of integration objects which could cause Siebel Adapter to fail upon update.

To deactivate primary fields

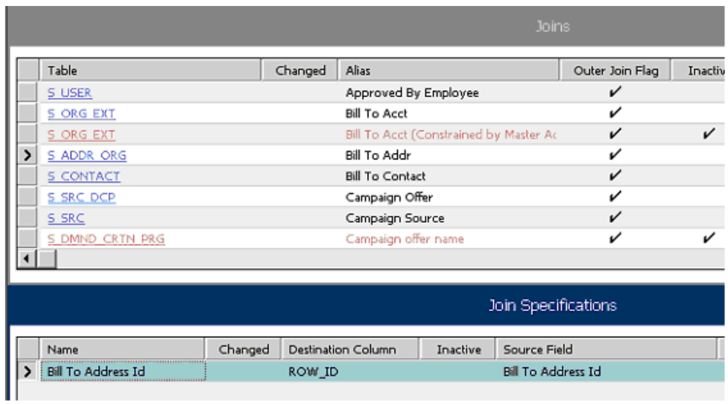
1. Determine which fields are primary.

   NOTE: Primary fields are usually prefixed with Primary but this is not always true. To find out which fields are primary fields, look in the definition of Business Components. The following example for BusComp Order Entry - Orders illustrates how to do this.

   ---

2. Within Siebel Tools, locate BusComp Order Entry - Orders.
3. Navigate to Business Component/Multi Value Link.
4. Record the Primary Id Field properties:
   - Account Address Id
   - Bill To Address Id
   - Bill To Contact Id
   - Approved By – Position Id
   - Primary Organization Id
   - Primary Position Id
   - Ship To Address Id
   - Ship To Contact Id

5 Navigate to Business Component/Join for Order Entry - Orders. Find the aliases for all the joins that have a join specification with Source Field property being one of the Ids. For example, join “Bill To Addr” has a join specification Bill To Address Id. The Source Field is Bill To Address Id.

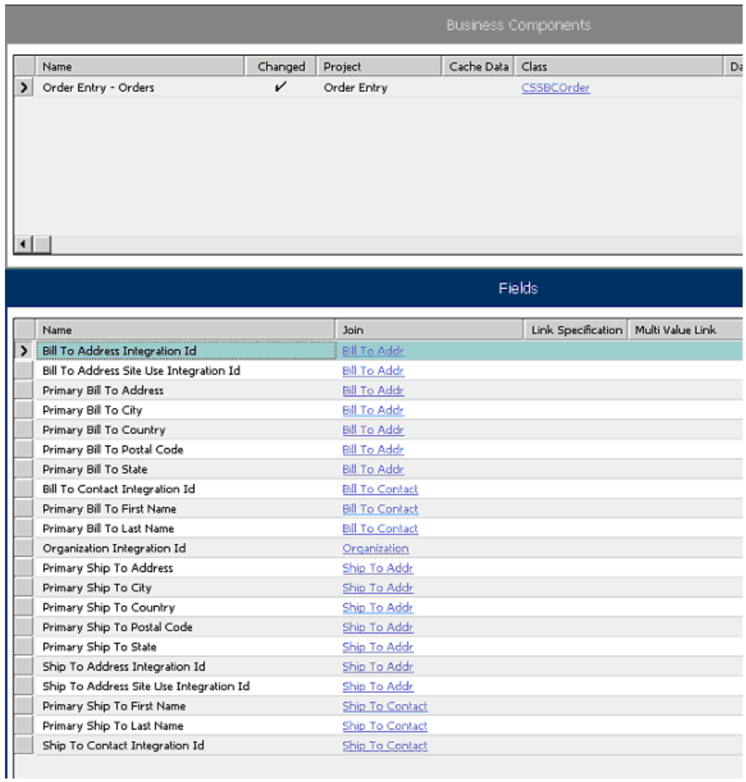


Following is a list of joins that result in Primary fields:

- Bill To Addr
- Bill To Contact
- Organization
- Primary Position
- Ship To Addr
- Ship To Contact

NOTE: Some Primary Id Fields do not have corresponding join specifications.

6 Navigate to Business Component/Field for Order Entry – Orders. Search for fields with a “Join” property.



7 Deactivate the Primary Id fields of the corresponding integration component.

8 Repeat for each integration component.

Adding SnapshotObjectType System Field

For each integration object, you must add a field called SnapshotObjectType for the root component. The field value specifies the incremental type (by component or by object). This field is used internally by the Siebel Connector for Siebel eBusiness Applications.

To add SnapshotObjectType field to root component

1. Set the external name to SnapshotObjectType.
2. Identify the XML sequence as any unique integer.
3. Set XML tag to SnapshotObjectType.
4. Set XML style to Attribute.
5. Set Type to System.

Adding RealOperation System Fields

For each integration component, you must add a field called RealOperation. The field value specifies the real operation: upsert, delete, or none. This field is used internally by the Siebel Connector for Siebel eBusiness Applications.

To add RealOperation field to integration components

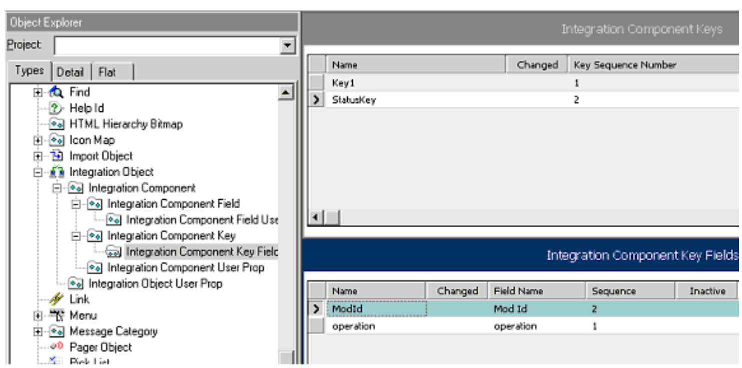
1. Set the external name to RealOperation.
2. Identify the XML sequence as any unique integer.
3. Set XML tag to RealOperation.
4. Set XML style to Attribute.
5. Set Type to System.

Adding a Status Key

For each integration component, add another Integration Component Key called StatusKey. Status Key is a meta-description of the data structure Siebel Adapter should return upon completion of execution.

To add a status key

1. Start Siebel Tools.
2. Navigate to Integration Objects > Integration Components > Integration Component Key.
3. Add a new record under Integration Keys with the following properties:
   - Name = StatusKey
   - Key Type = StatusKey
4. Navigate to Integration Objects > Integration Components > Integration Component Key > Integration Component Key Field.



5. Add the following new records:

Name	Field
operation	operation
ModId	Mod Id

6 Add any other Mod Id field defined from the extension tables of the base table.

7 Add a Component Key Field to Status Key with the following properties:

- Name = S-S Key Id
- Value = S-S Key Id

Adding Field Dependency

For each integration component, add a user property FieldDependencyInt to the field S-S Key Id. This instructs Siebel Adapter to evaluate S-S Instance before S-S Key Id. The field S-S Key Id is dependent on field S-S Instance.

To add field dependency to the S-S Key ID

- Set the Value field for FieldDependencyInt to S-S Instance.

Adding User Property to Allow Updates

Global records can not be updated or deleted unless by their owner instances. However, Siebel Adapter must have the ability to perform these operations on behalf of the record's owner instance.

To add user property to allow updates

- Set the ForcedUpdate user property to Y at the Integration Object level.

Add Other User Keys (Except for Status Key)

For each integration component defined, create a user key that contains only S-S Key Id field. All user keys other than status key and the user key containing S-S Key Id should be deleted. This is because Siebel adapter uses only one user key for insert or update. Having more than one user key may cause unexpected behavior.

Customizing a DTE Map for Use with Siebel Connector

For any data map created using the Siebel Data Mapper to work with the Siebel Connector for eBusiness Applications, you must first modify the map, as explained below.

To modify a data map for use with Siebel Connector

1. Navigate to the Integration Administration view.
2. From there, navigate to Data Map Editor.
3. Locate the DTE map you want to use.
4. On the Integration Component Map applet, set Source Search Specification to `[operation]<>"none"` for the root component.
5. On the Integration Component Map applet, create a key component for the root component.
6. Set Source Search Specification to `[operation]="none"`.
7. On the Integration Field Map applet, create the following fields for the key root component.

Source Expression	Target Field Name
"upsert"	Operation
[S-S Key Id]	S-S Key Id
[SnapshotObjectType]	SnapshotObjectType
[RealOperation]	RealOperation

NOTE: See *Business Processes and Rules: Siebel eBusiness Application Integration Volume IV* for information on how to create DTE maps using the Siebel Data Mapper.

Initializing a New Instance 5

This chapter describes how to implement *Siebel Connector for Siebel eBusiness Applications* when you are connecting new instances to an existing instance. This includes creating and adding the new instances to establish a Siebel Connector deployment. This section also describes how to copy global data from the original instance and propagate it to the newly created instances.

Checklist

- ❑ Create new instances.

  For details, see "Creating New Instances" on page 57.

- ❑ Create a publication group.

  For details, see "Creating a Publication Group" on page 24.

- ❑ Create a channel and associate integration IDs with channels.

  For details, see "Creating a Channel" on page 28 and "Associating Integration Objects with Channels" on page 29.

- ❑ Identify the records you want to share and assign ownership of a record.

  For details, see "Identifying Records to Share" on page 58 and "Assigning Ownership of a Record to an Instance" on page 58.

- ❑ Create a database copy.

  For details, see "Create a Copy of the Database" on page 58.

- ❑ Update instance IDs.

  For details, see "Update Instance Identifications" on page 58.

- ❑ Modify channels.

  For details, see "Modifying Channel Information" on page 59.

- Run workflows.

  For details, see "Run the Workflow Processes" on page 60.

Creating New Instances

This section describes how to create a new instance of Siebel.

To create a new instance of Siebel

1. Determine a Major ID and Minor ID for the new instance.

   NOTE: When selecting an appropriate Major Segment, Siebel recommends using the DUNS # relevant for the instance. For the Minor Segment, use something descriptive or informative.

   ---

2. Navigate to the Siebel Instances view.

   Click New.

3. Fill out the form.
4. Click Save.
5. Repeat this procedure for as many instances you need to create for the adjusted deployment.

Publication Groups

For information on creating publication groups, see "Creating a Publication Group" on page 24.

Channels

For information on creating channels, see "Creating a Channel" on page 28.

Integration Objects

For information on associating integration objects with channels, see "Associating Integration Objects with Channels" on page 29.

Identifying Records to Share

Once you have created instances, you must prepare global data for them.

To identify records to become global data

1. Determine which records your organization wants to share within the deployment.
2. Review the list of records that will become global data and make sure information in these records is correct.

Assigning Ownership of a Record to an Instance

You must set the ownership of records you want to share to the initial instance in your deployment. Setting ownership of records for global data is a manual process.

To set ownership of records to the initial instance

1. Navigate to the view for the record you want to designate as global data.
2. Click Edit and set Owner Instance field to the instance ID of the initial instance.
3. Repeat this procedure for each record you want to become global data.

NOTE: For additional details regarding ownership of global data, see "Global Data" on page 15.

Create a Copy of the Database

You need to create a copy of the database on the initial instance of Siebel Applications in preparation for sending it to the new instances.

Update Instance Identifications

For information on updating instance IDs, see "Defining the HQ Instance ID" on page 27 and "Creating a Channel" on page 28.

Modifying Channel Information

You must make adjustments to the channel information after you add new instances to correctly identify the publishing and receiving queues. For example, a receiving instance (A) may become the publishing machine. Therefore, the publishing queue should point to instance (A). A new instance (B) may become the receiving machine. The receiving queue should point to the new instance (B).

To add remaining node information

1. Navigate to the Channels view and select a channel requiring modification.
2. Click Edit.

   The Channels screen appears.
3. Modify the information as required.
4. Click Save.

Repeat this procedure for any other channels requiring modification.

Run the Workflow Processes

The final phase of this scenario is to run the publishing and receiving workflow processes. See "Publishing Messages" on page 39 and "Receiving Messages" on page 40 to generally learn about these processes. The following sections provide detailed information.

The general workflow for these processes is shown in Figure 5.

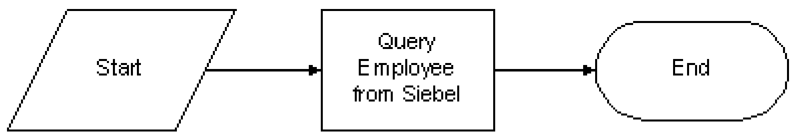


Figure 5. General Workflow Process

Siebel to Siebel Outbound Process

The Siebel to Siebel (or S2S) Outbound Process is based on business object Channel Object. The Workflow Process tab display for the S2S Outbound Process is shown in Figure 6.

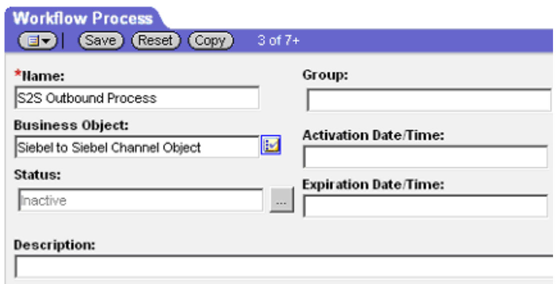


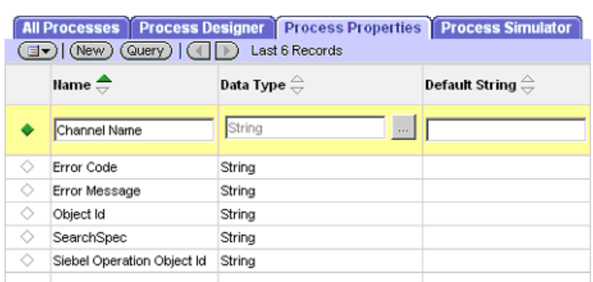
Figure 6. Siebel to Siebel Outbound Process Workflow Process Tab

Process Properties

Channel Name: The Channel Name to which this channel object belongs. This parameter is taken from the ChannelName field of business component Siebel to Siebel Channel.

SearchSpec: The Search Specification to filter global data. This property is taken from the Search Specification field of business component Channel Object.

All other properties are workflow default process properties.

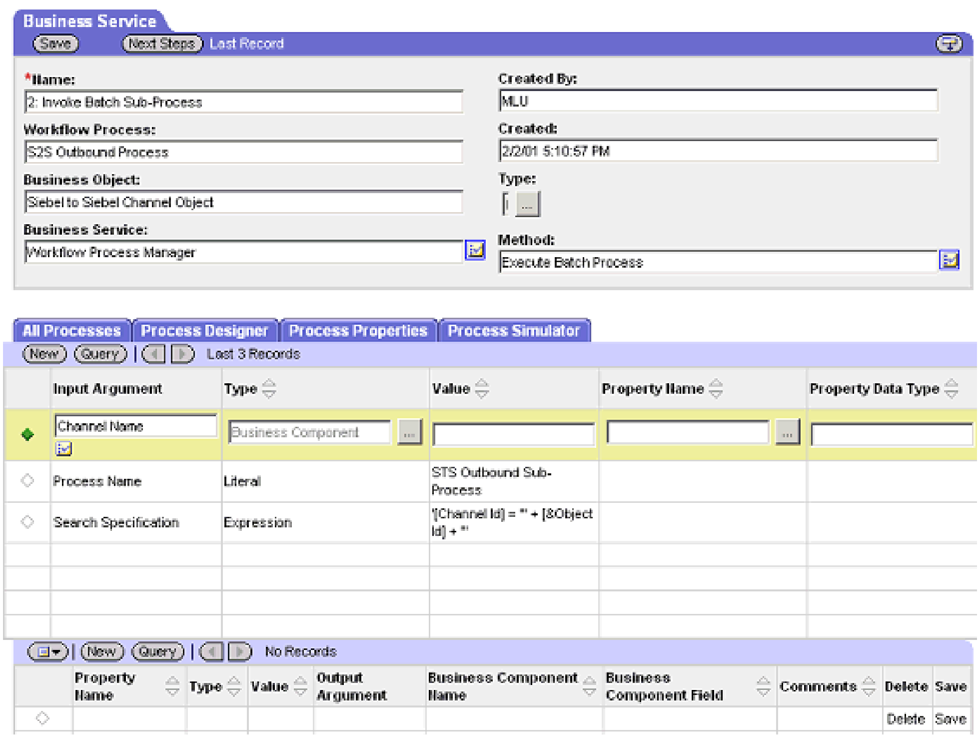


Process Steps

1 Start.

2 Invoke Batch Sub-Process.

This step invokes the Execute Batch Process method of business service Workflow Process Manager. S2S Outbound Sub-Process is invoked on each channel object that satisfies Search Specification [Channel Id] = [Object Id].



Siebel to Siebel Outbound Sub-Process

This is the main publishing workflow. It operates on each channel object. The corresponding business object for this workflow (shown in Figure 7) is the Siebel to Siebel Object.

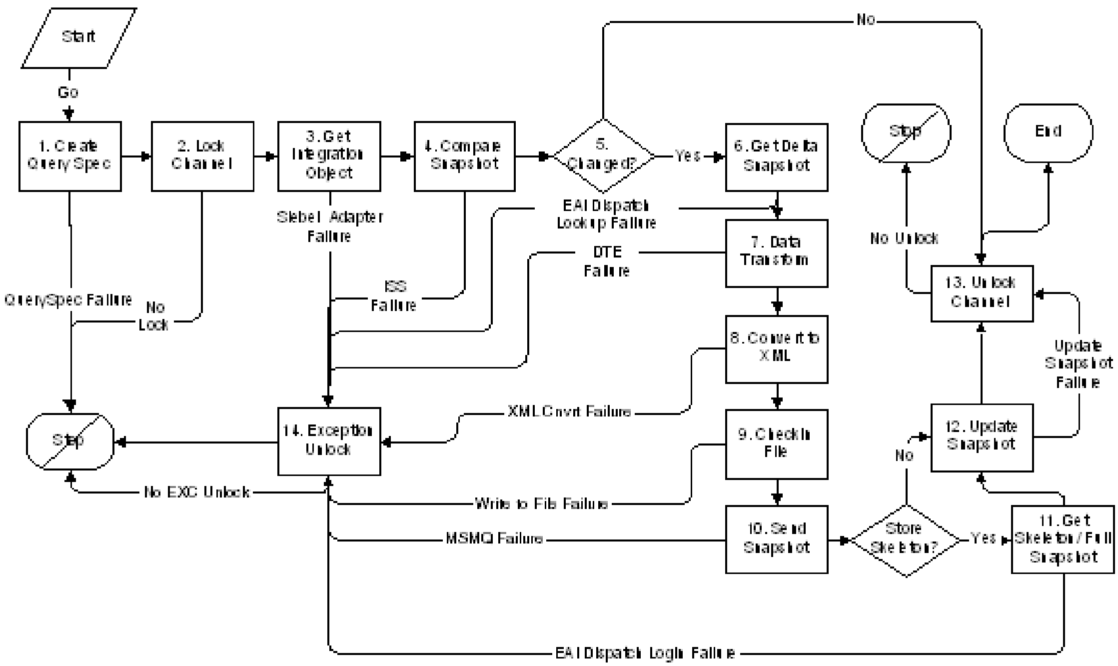


Figure 7. Siebel to Siebel Outbound Sub-Process Workflow

Process Properties

Used internally in this workflow process, Figure 8 through Figure 11 on page 65 show the process rows displayed under the Process Properties tab.

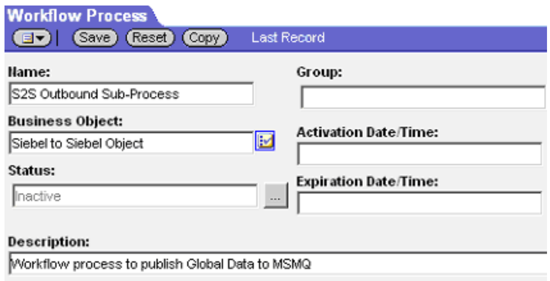
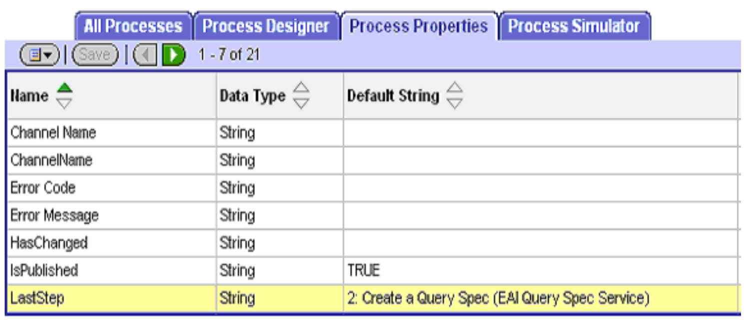
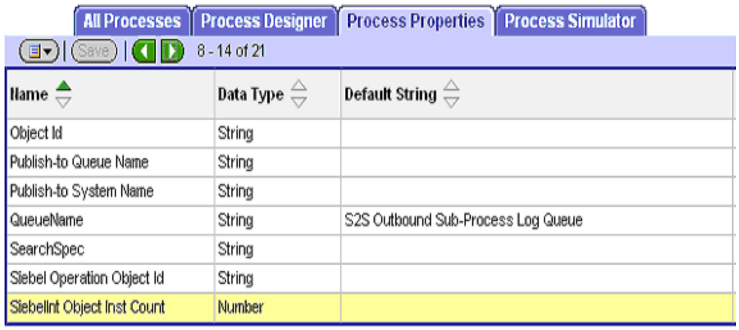


Figure 8. Process Properties for Outbound Sub-Process Workflow (1 of 4)



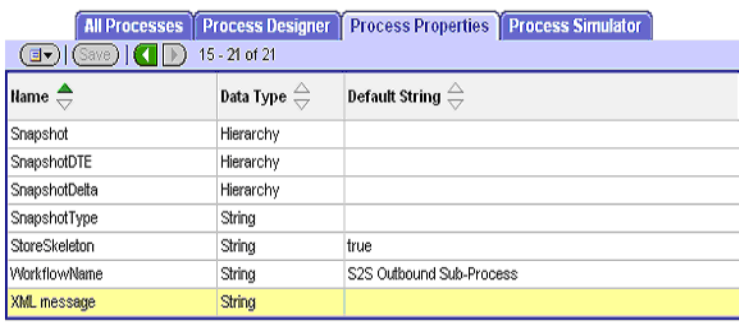
Name	Data Type	Default String
Channel Name	String	
ChannelName	String	
Error Code	String	
Error Message	String	
HasChanged	String	
IsPublished	String	TRUE
LastStep	String	2: Create a Query Spec (EAI Query Spec Service)

Figure 9. Process Properties for Outbound Sub-Process Workflow (2 of 4)



Name	Data Type	Default String
Object Id	String	
Publish-to Queue Name	String	
Publish-to System Name	String	
QueueName	String	S2S Outbound Sub-Process Log Queue
SearchSpec	String	
Siebel Operation Object Id	String	
SiebelInt Object Inst Count	Number	

Figure 10. Process Properties for Outbound Sub-Process Workflow (3 of 4)

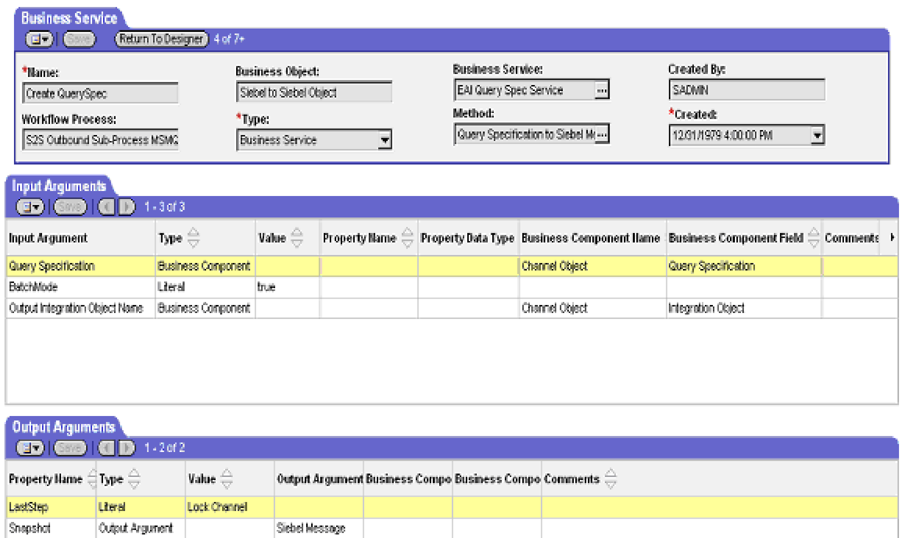


Name	Data Type	Default String
Snapshot	Hierarchy	
SnapshotDTE	Hierarchy	
SnapshotDelta	Hierarchy	
SnapshotType	String	
StoreSkeleton	String	true
WorkflowName	String	S2S Outbound Sub-Process
XML message	String	

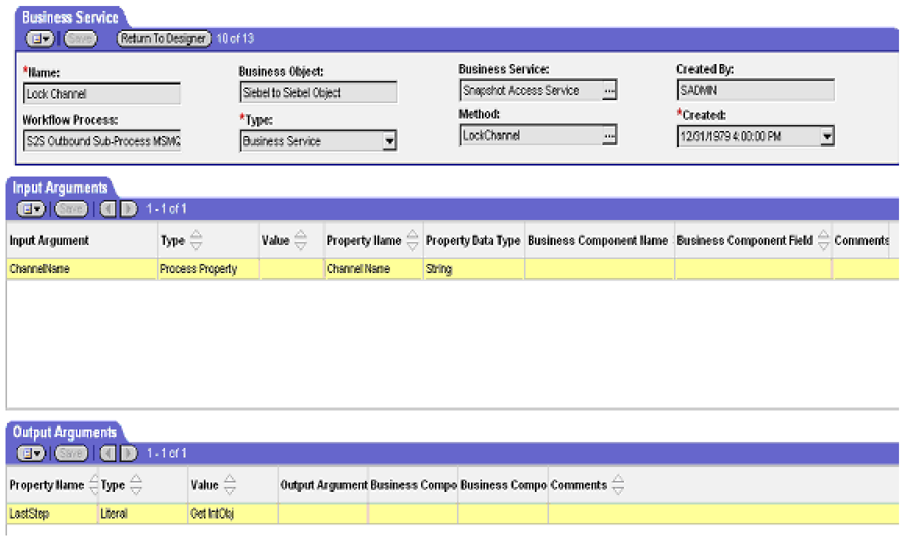
Figure 11. Process Properties for Outbound Sub-Process Workflow (4 of 4)

Process Steps

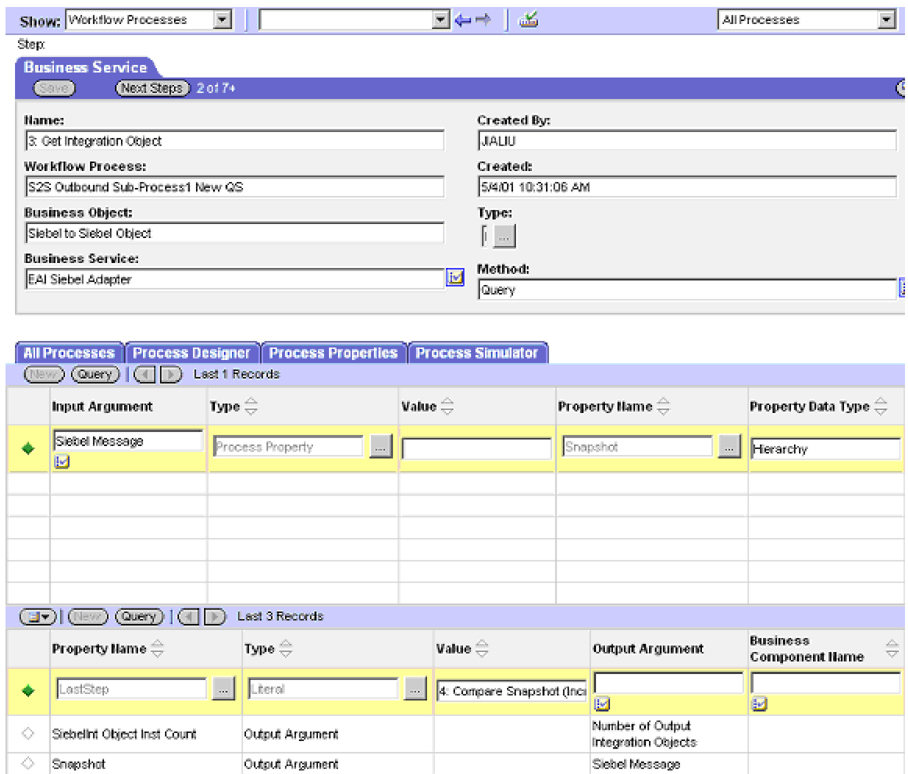
1 **Create a Query Spec:** This step invokes the Query Specification to Siebel Message method of business service EAI Query Spec Service. This service builds an intermediate data structure that satisfies the specified Query Spec and integration object definition.



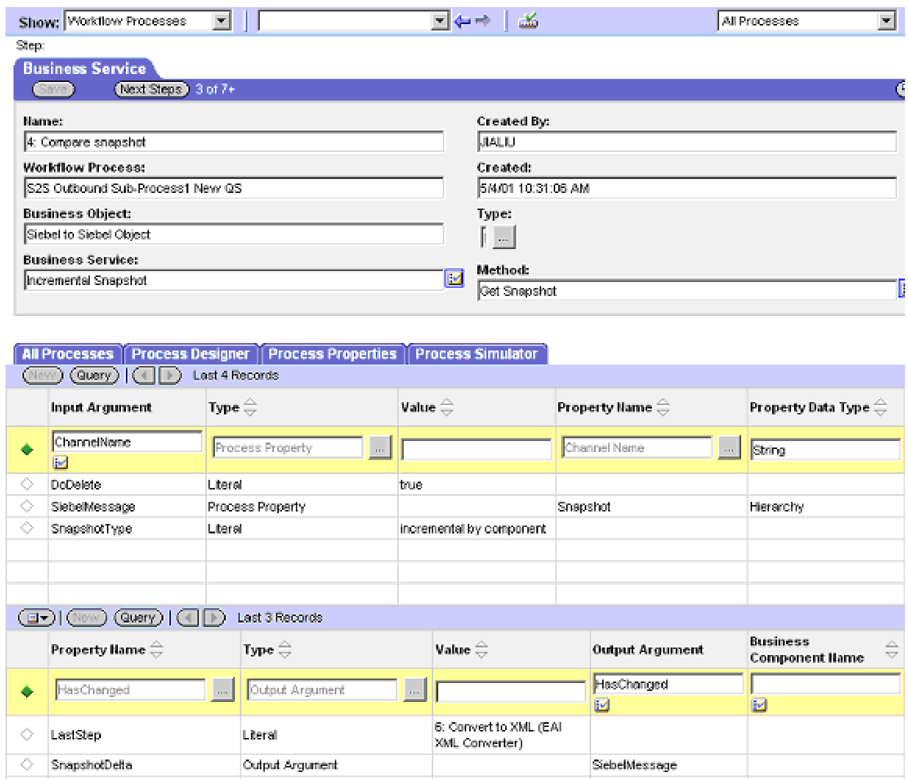
2 **Lock Channel:** This step invokes the LockChannel method of Business Service Snapshot Access Service. It is used to synchronize the publishing process and receiving process on the same instance for a given channel.



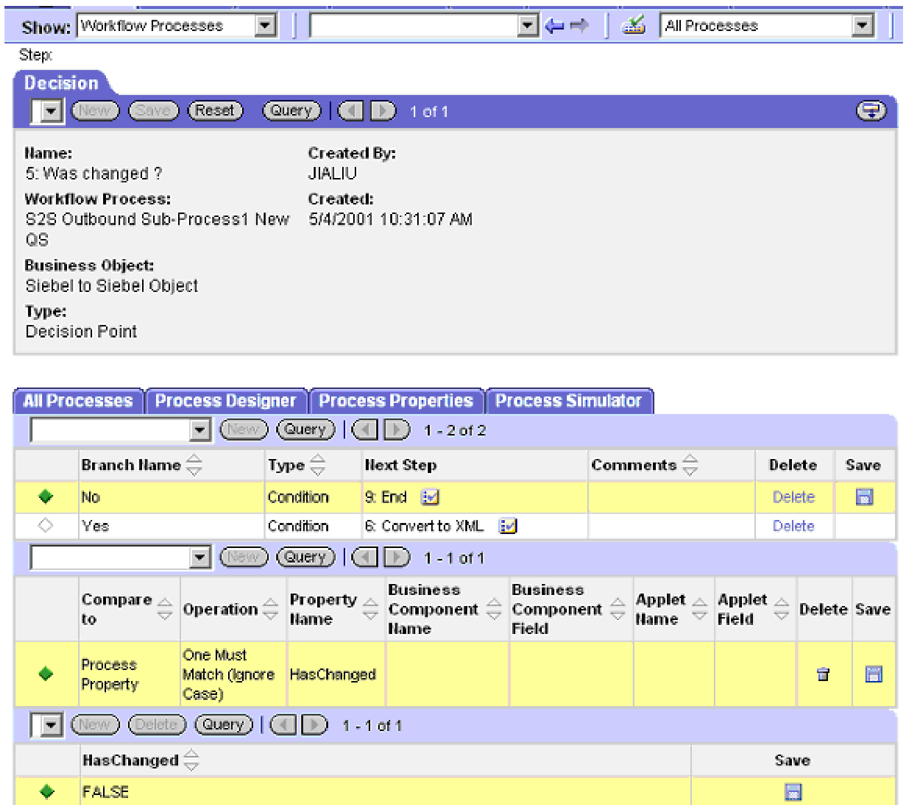
3 **Get Integration Object:** This step invokes the Query method of Business Service EAI Siebel Adapter. Siebel Adapter takes the output of Step 2 and builds a data structure that contains all the integration object instances that satisfy the query specification.



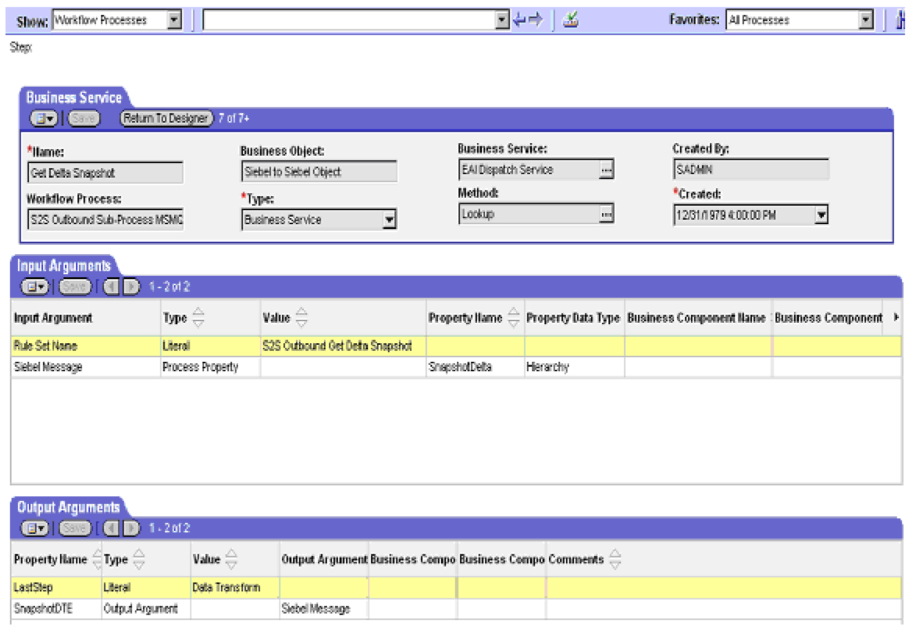
4 **Compare Snapshot:** This step invokes the Get Snapshot method of business service Incremental Snapshot. Incremental Snapshot takes the integration object instances that were queried from Siebel Adapter and compares them with what has been taken in the snapshot table. Only the differences (a partial image of integration object instances) are output (to “SnapshotDelta”).



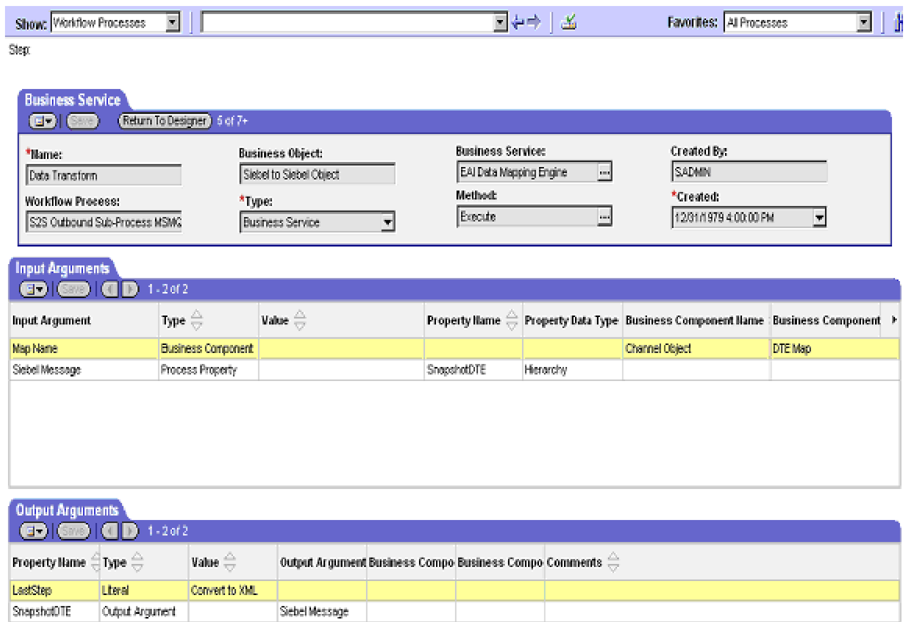
5 **Was changed?:** This is a decision step. If the current global data image is different from what is in the snapshot table, go to Step 6: Convert to XML; otherwise, this workflow ends.



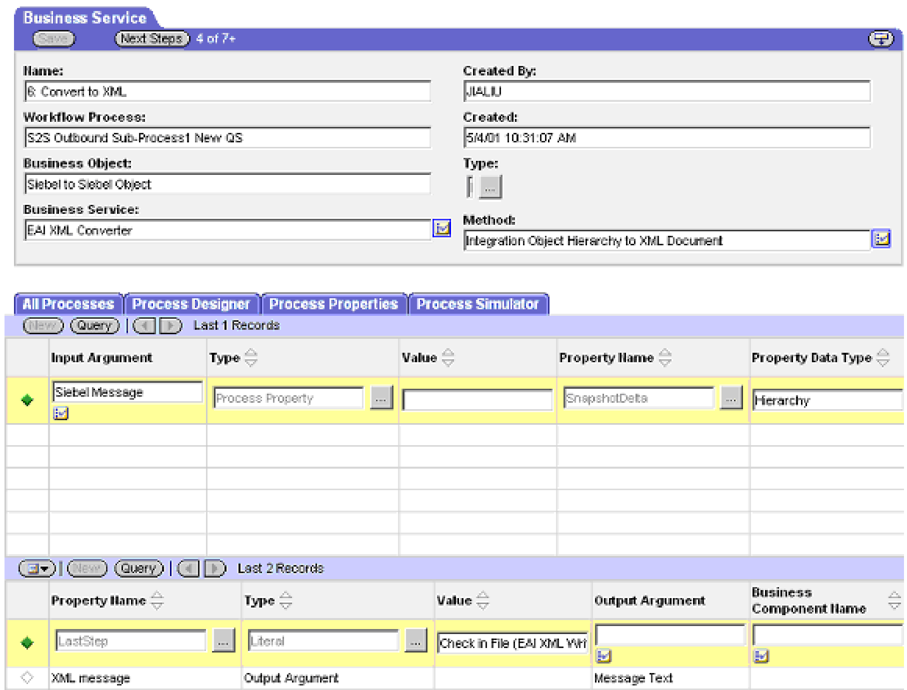
6 **Get Delta Snapshot:** This step invokes the Lookup method of business service EAI Dispatch Service. The output of Incremental Snapshot has two Siebel Messages. One is the calculated change and the other is the skeleton/full image for update snapshot. EAI Dispatch Service gets the calculated change from the outputs at this step.



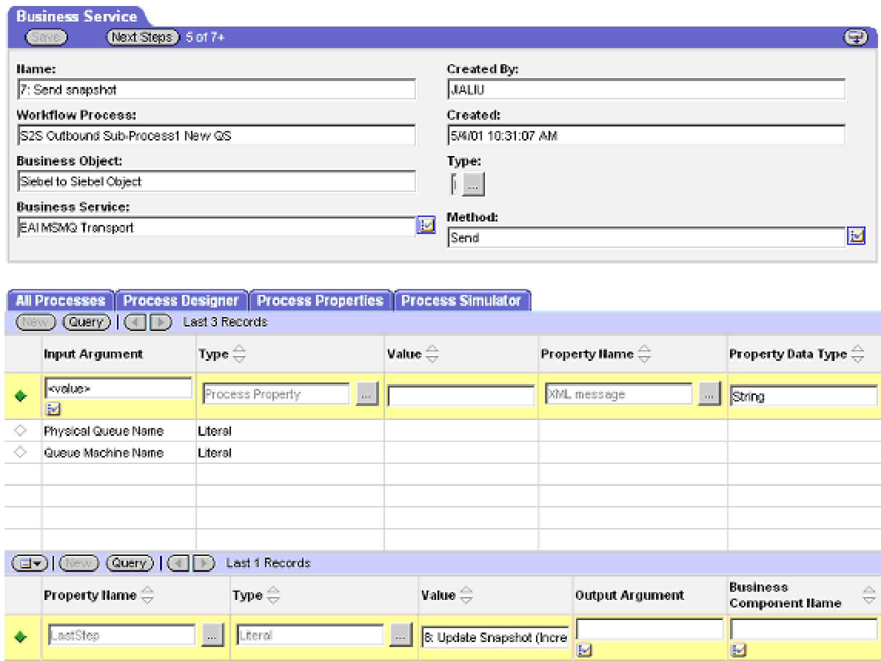
7 **Data Transformation:** This step invokes the Execute method of business service EAI Data Mapping Engine. The Engine transforms the integration object instance to the version that used on the receiving side.



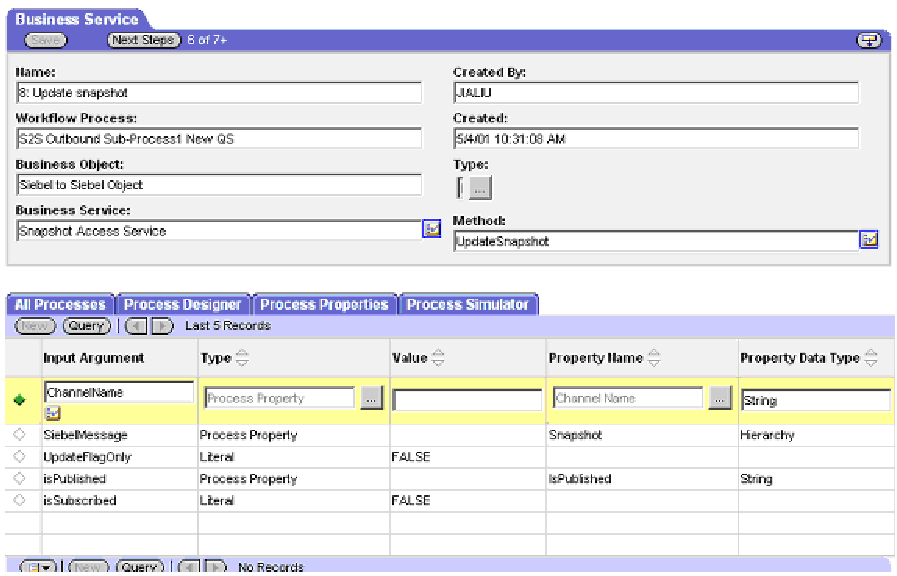
8 **Convert to XML:** If the current global data image is different from what is in the snapshot table, the difference (contained in SnapshotDelta) is converted to an XML document using the Integration Object Hierarchy to XML Document method of business service EAI XML Converter.



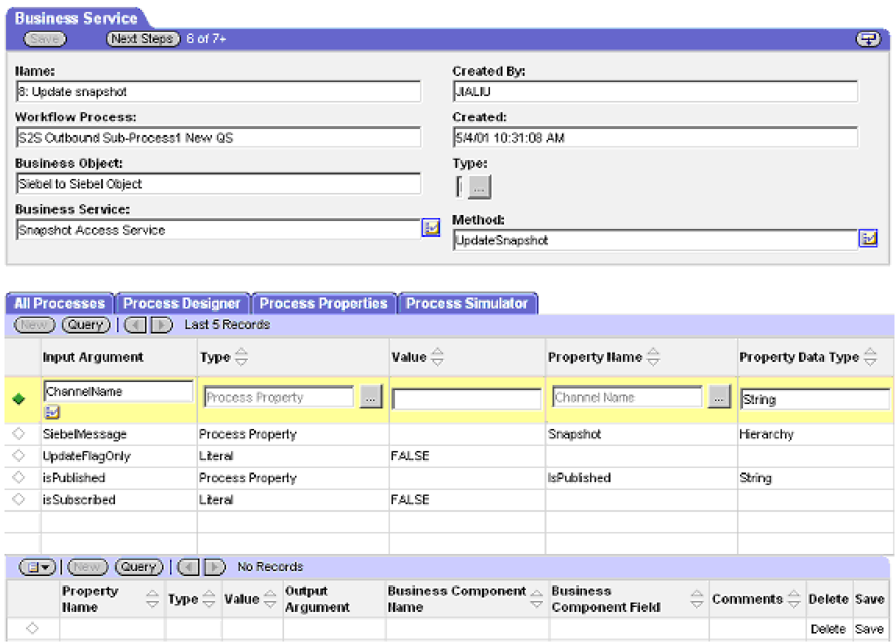
9 **Send Snapshot:** The converted XML Document is sent to a specific queue on a specific machine by EAI MSMQ Transport business service.



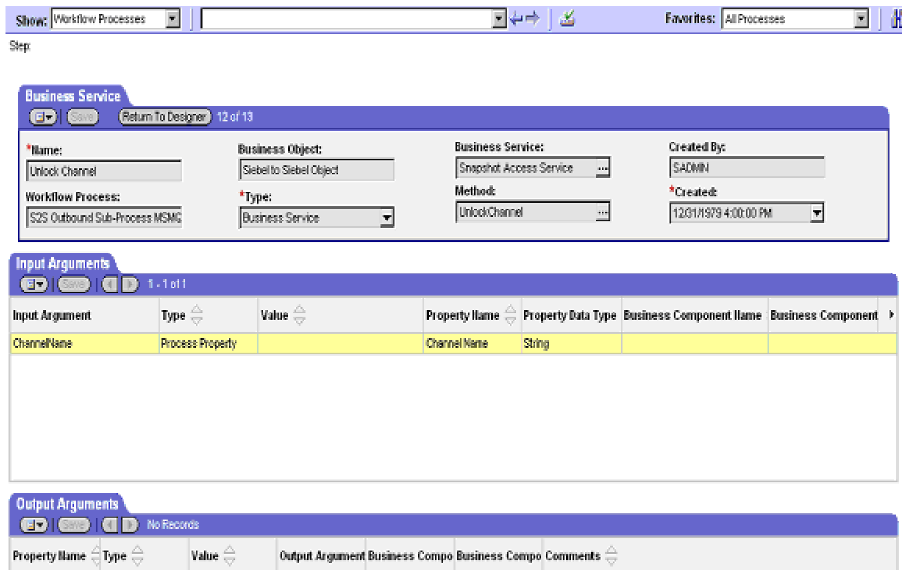
10 **Get Skeleton/Full Snapshot:** This step invokes the Lookup method of business service EAI Dispatch Service. EAI Dispatch Service gets the skeleton/full snapshot image and passes it as the input of "Update Snapshot."



11 **Update Snapshot:** After the incremental snapshot is sent, the snapshot table is updated to reflect the current image of global data (thus, if global data is modified in the future, only the difference in the new image and current image will be published next time).



12 **Unlock Channel:** This step invokes the UnlockChannel method of business Service Snapshot Access Service. It releases the lock on the channel so that the receiving process can start to process data. It is used to synchronize the publishing process and receiving process on the same instance for a given channel.



13 **Exception Unlock:** Calls the same method as the "Unlock Channel." If a step fails before the end of the workflow, the workflow releases the lock so that the receiving process can proceed.

Siebel to Siebel Inbound Process

The Siebel to Siebel (or S2S) inbound process is the main receiving workflow that operates on a channel object. The corresponding business object for this workflow is Siebel to a Siebel Channel Object. The workflow is shown in Figure 12.

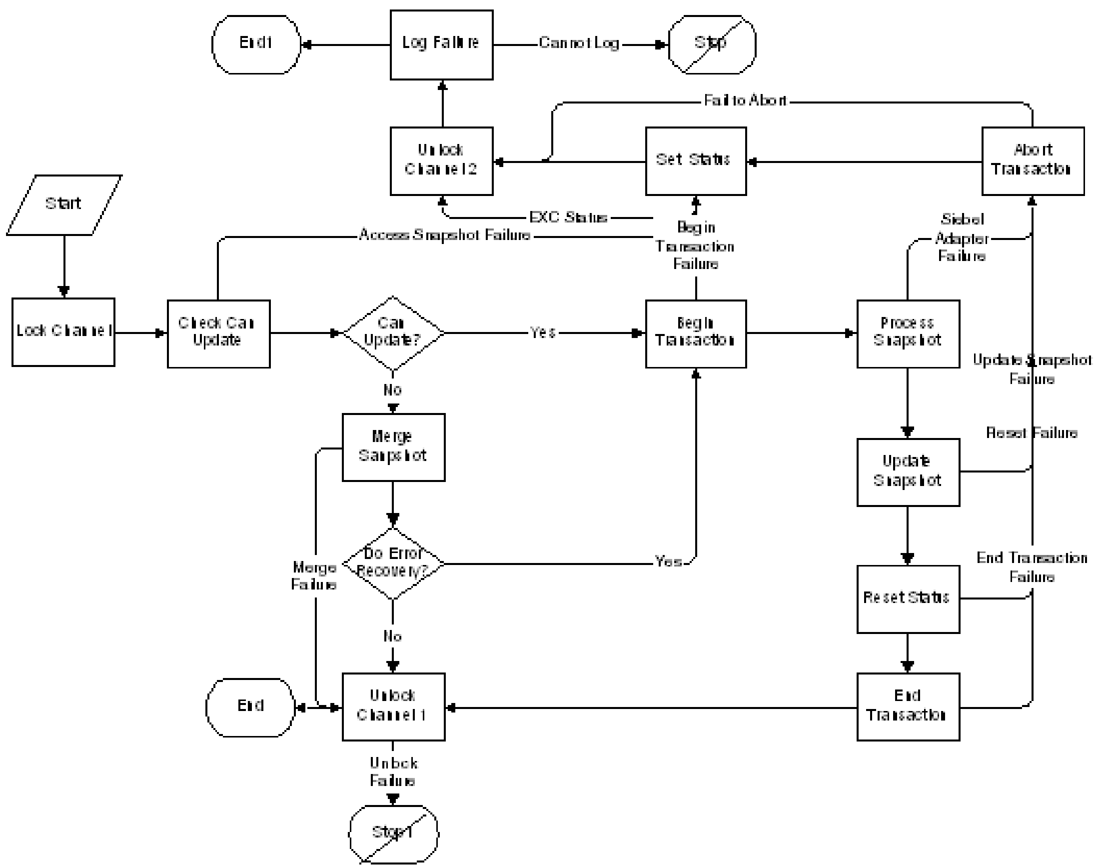


Figure 12. Siebel to Siebel Inbound Process Workflow

The Workflow Process tab display for the S2S Inbound Process is shown in Figure 13.

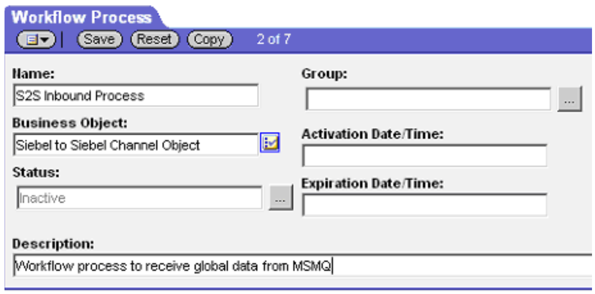


Figure 13. Siebel to Siebel Inbound Process Workflow Process Tab

Process Properties

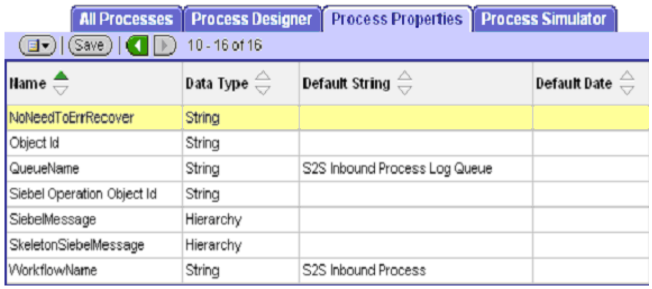
Used internally in this workflow process, Figure 14 and Figure 15 show the process rows displayed under the Process Properties tab.

Name	Data Type	Default String	Default Date
CanUpdate	String		
Channel Name	String	Channel BC	
DeleteRoot	String		
Error Code	String		
Error Message	String		
ForceToErrRecover	String		
KeyFieldName	String		

Figure 14. Process Properties for Inbound Process Workflow (1 of 3)

Name	Data Type	Default String
LastStep	String	Lock Channel
Object Id	String	
QueueName	String	S2S Inbound Process Log Queue
Siebel Operation Object Id	String	
SiebelMessage	Hierarchy	
SkeletonSiebelMessage	Hierarchy	
WorkflowName	String	S2S Inbound Process

Figure 15. Process Properties for Inbound Process Workflow (2 of 3)

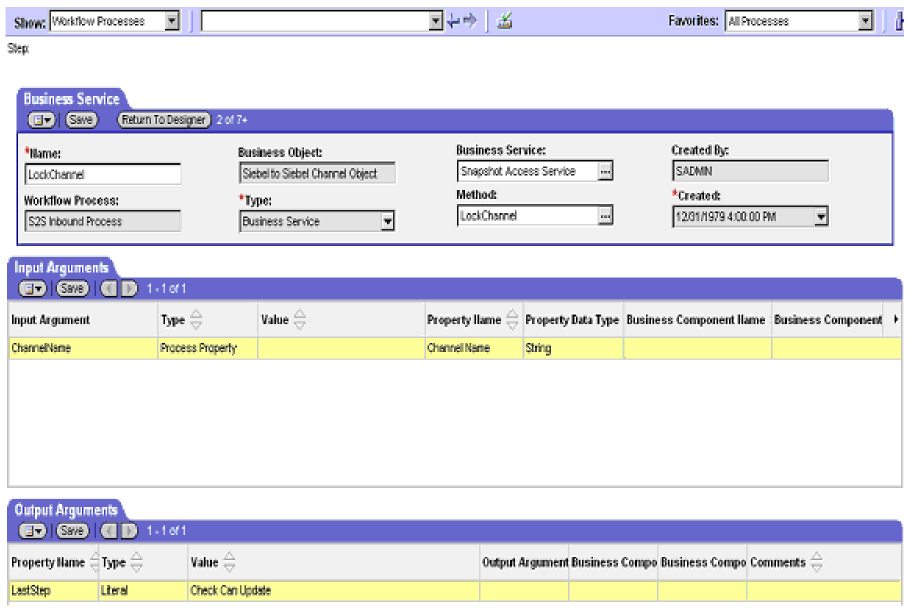


Name	Data Type	Default String	Default Date
NoNeedToErrRecover	String		
Object Id	String		
QueueName	String	S2S Inbound Process Log Queue	
Siebel Operation Object Id	String		
SiebelMessage	Hierarchy		
SkeletonSiebelMessage	Hierarchy		
WorkflowName	String	S2S Inbound Process	

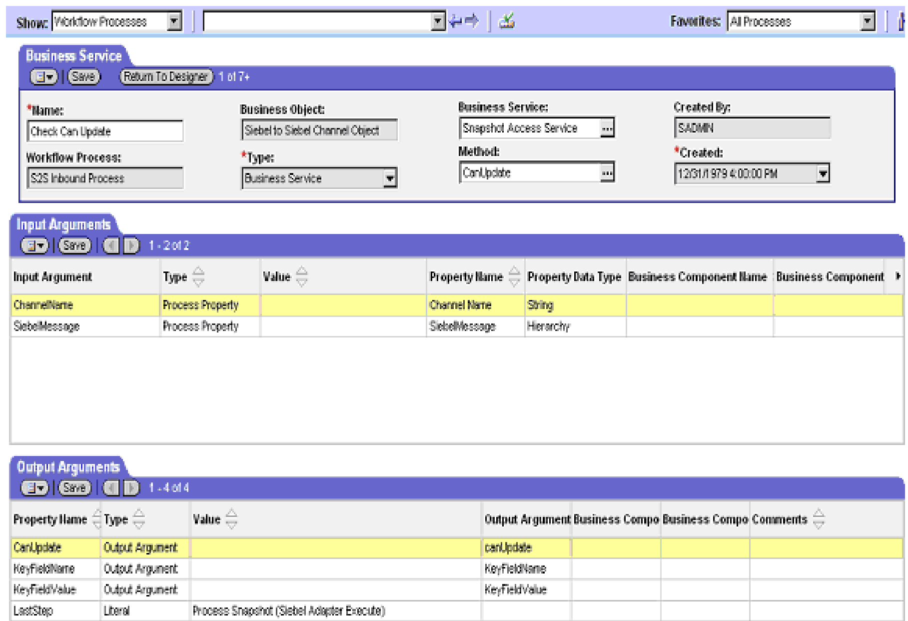
Figure 16. Process Properties for Inbound Process Workflow (3 of 3)

Process Steps

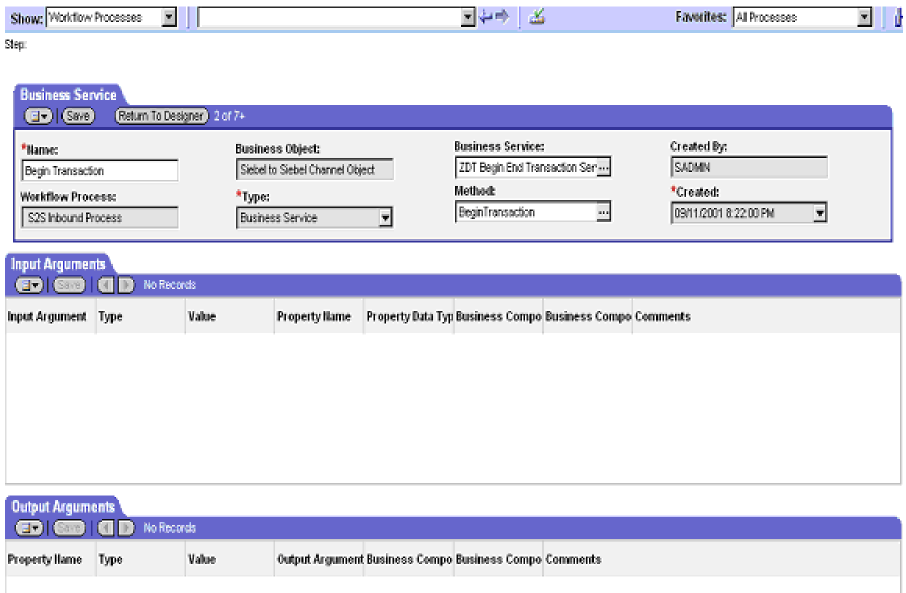
1 **Lock Channel:** This step invokes the LockChannel method of business service Snapshot Access Service. It is used to synchronize the publishing process and receiving process on the same instance for a given channel.



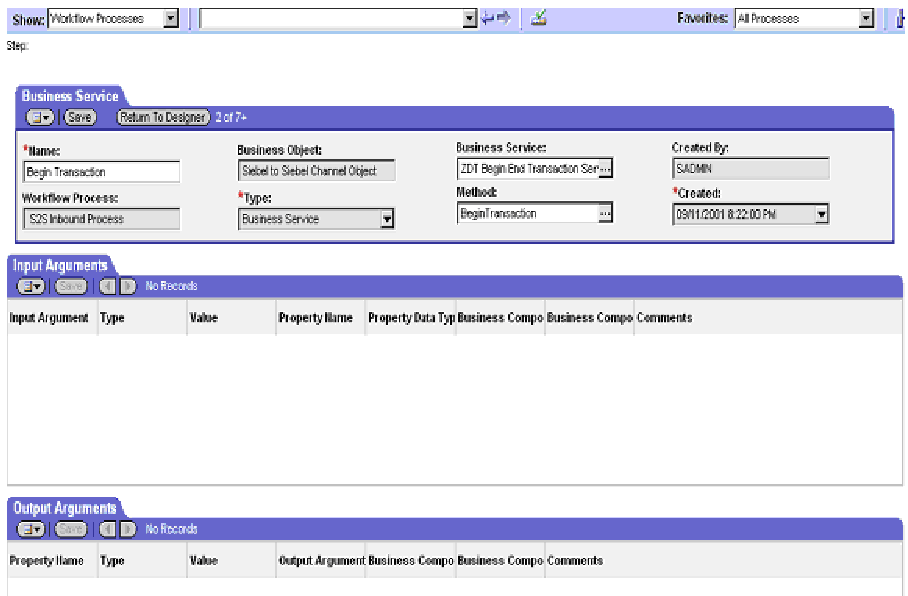
2 **Check Can Update:** This step checks the validity of the input Siebel message and returns a parameter (canUpdate) that indicates whether the snapshot should be updated.



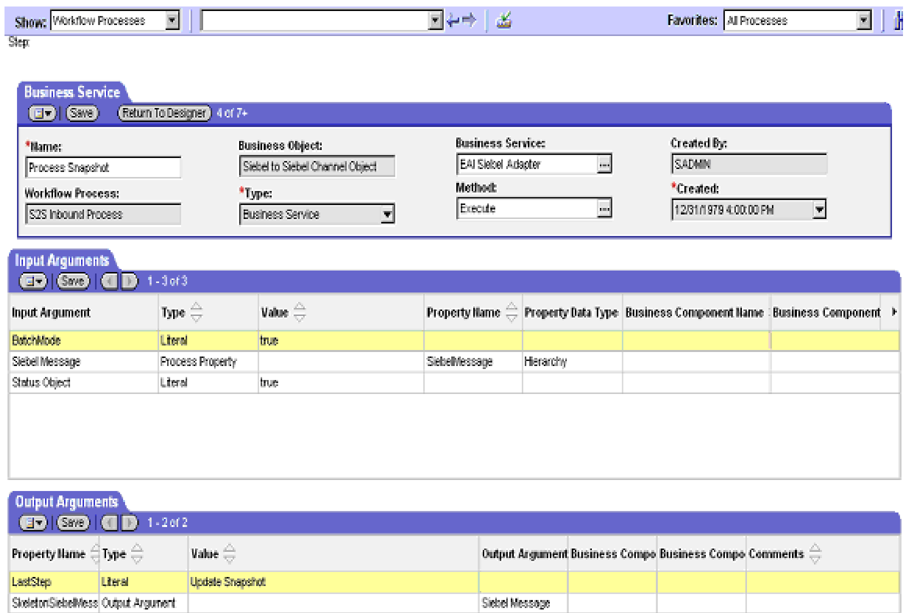
3 **Can Update?:** The output of Step 2: Access Snapshot determines whether the input message is valid. If true, go to Step 4: Process Snapshot; otherwise, this workflow process ends.



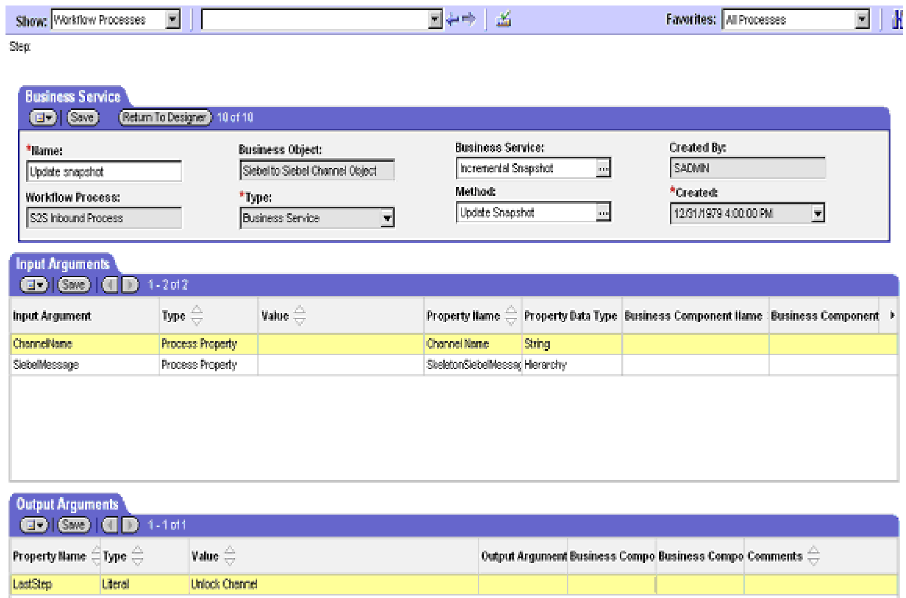
4 **Begin Transaction:** This step invokes the BeginTransaction method of business Service ZDT Begin End Transaction Service. It starts a transaction and makes certain that Process Snapshot, Update Snapshot, and Reset Status are in one transaction.



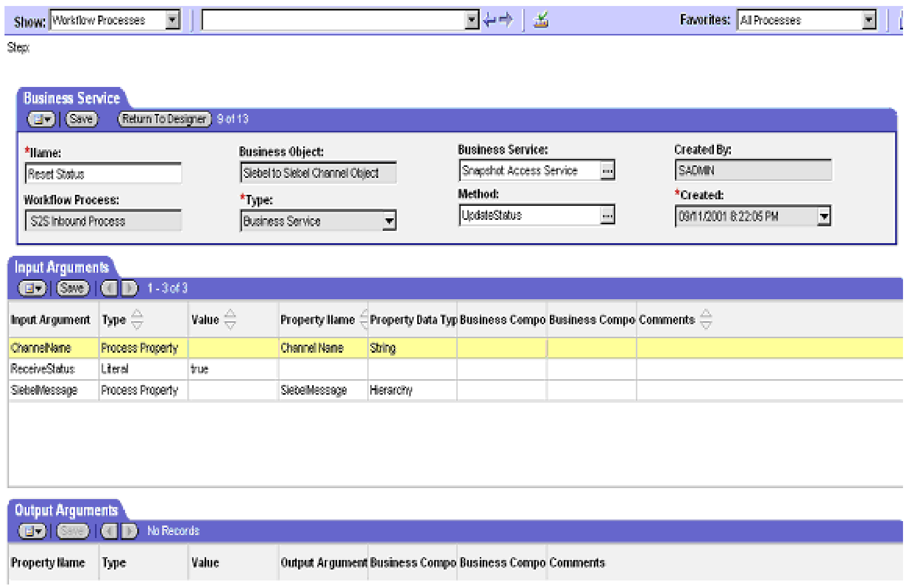
5 **Process Snapshot:** This step calls the Execute method of the AI Siebel Adapter business service to apply the changes that are contained in the incoming message to database.



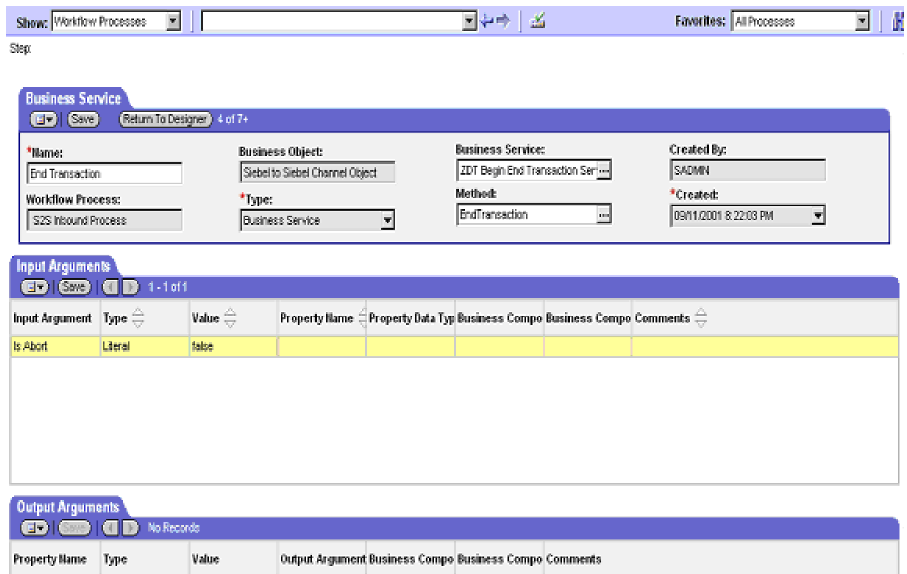
6 **Update Snapshot:** After Siebel Adapter applies the changes, the snapshot table is updated to record the new current images of this global data.



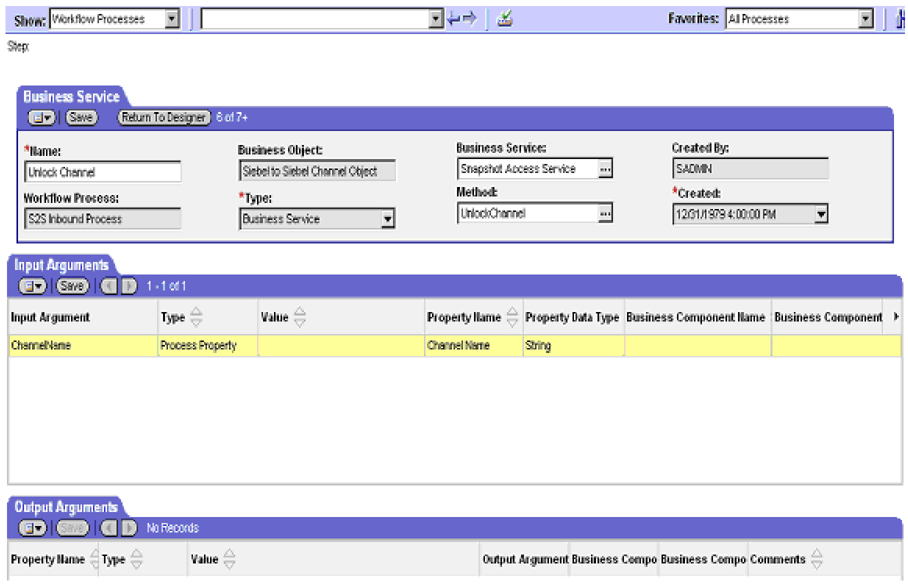
7 **Reset Status:** This step invokes the UpdateStatus method of the Snapshot Access Service business service. It sets ReceiveStatus to True if this is an error recovery case.



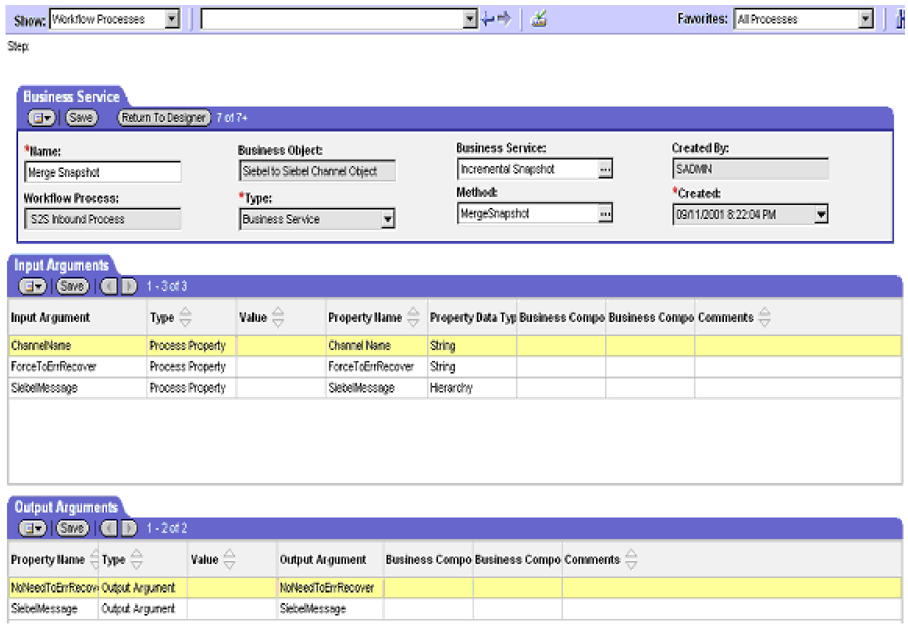
8 **End Transaction:** This step invokes the EndTransaction method of the ZDT Begin End Transaction Service business service. It ends a transaction and makes certain that Process Snapshot, Update Snapshot, and Reset Status are in one transaction.



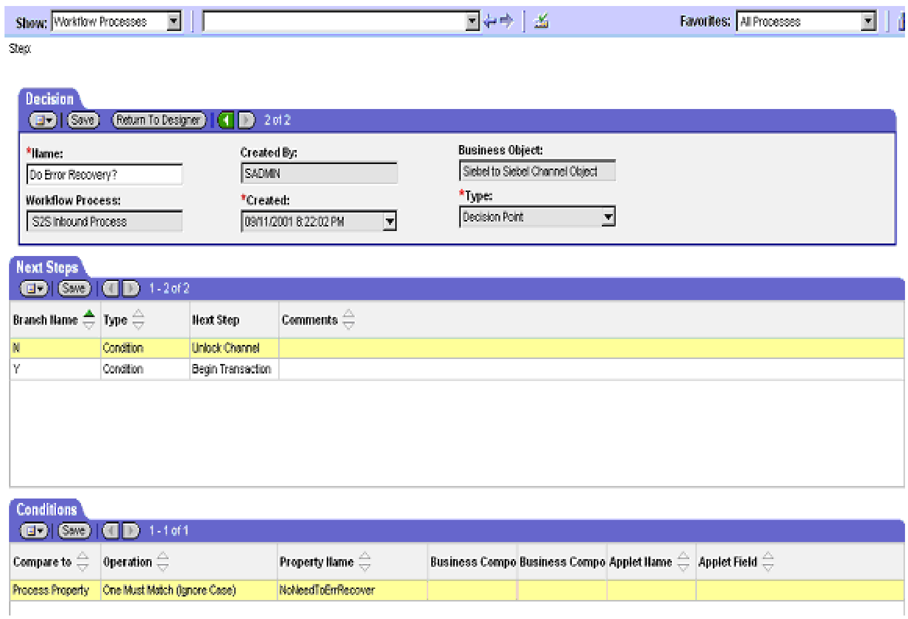
9 **Unlock Channel:** This step invokes the UnlockChannel method of business service Snapshot Access Service. It releases the lock on the channel so that the receiving process can start to process data. It is used to synchronize the publishing process and receiving process on the same instance for a given channel.



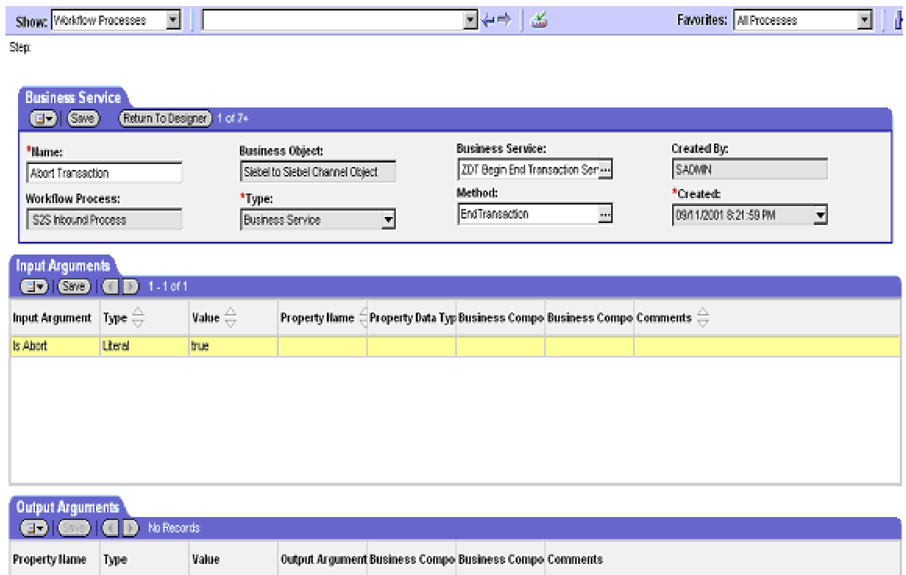
10 **Merge Snapshot:** This step invokes the MergeSnaphot method of the Incremental Snapshot business service. In case of error recovery, it merges the published changes together and stores it in the EAI Queue.



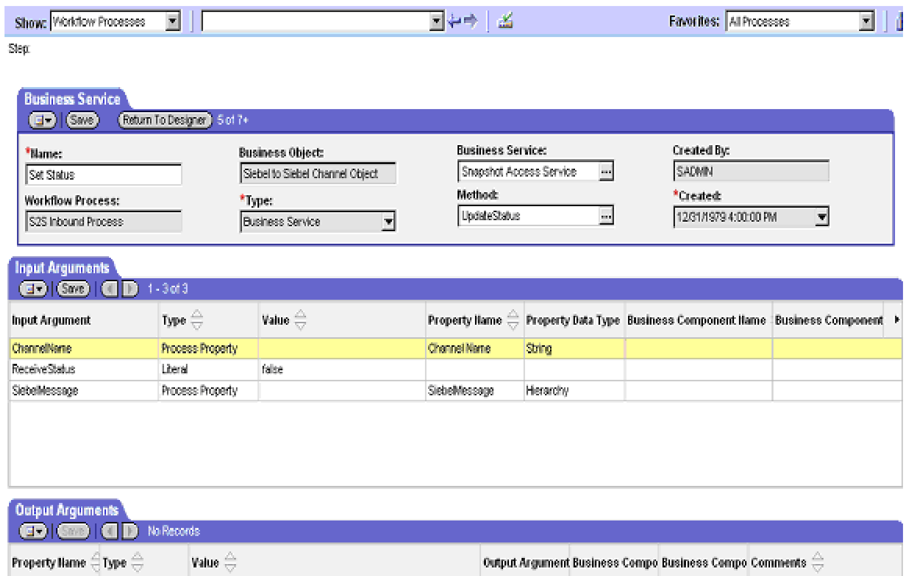
11 **Do Error Recovery?:** This is a decision point. If the current record is under the error recovery process and this is not the last publish to fix the error, it releases the lock and ends. If the current record is under the error recovery process and this is the last publish to fix the error, then go to the Begin Transaction step.



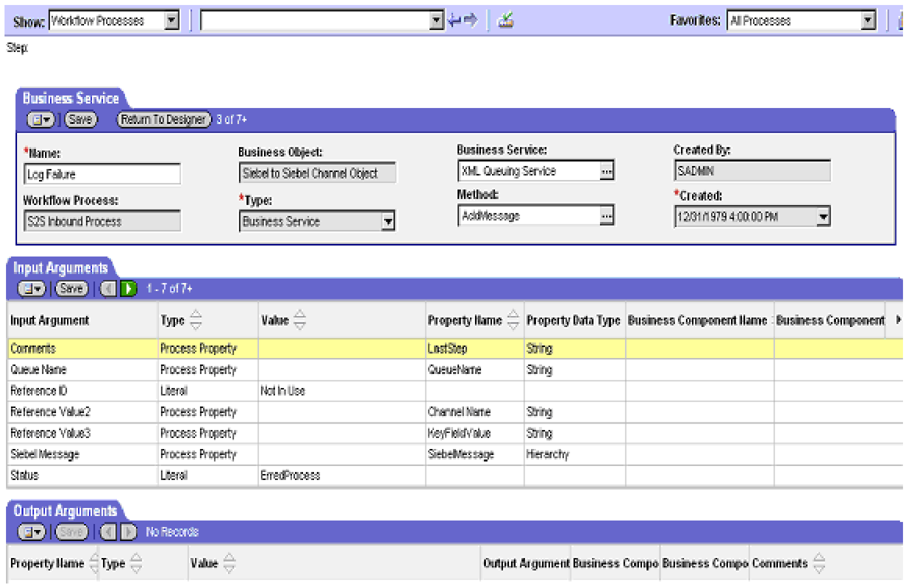
12 **Abort Transaction:** This step invokes the EndTransaction method of the ZDT Begin End Transaction Service business service. It ends the transaction and aborts any steps up to this point. If any step of Snapshot, Update Snapshot, or Reset Status fails, this step is called.



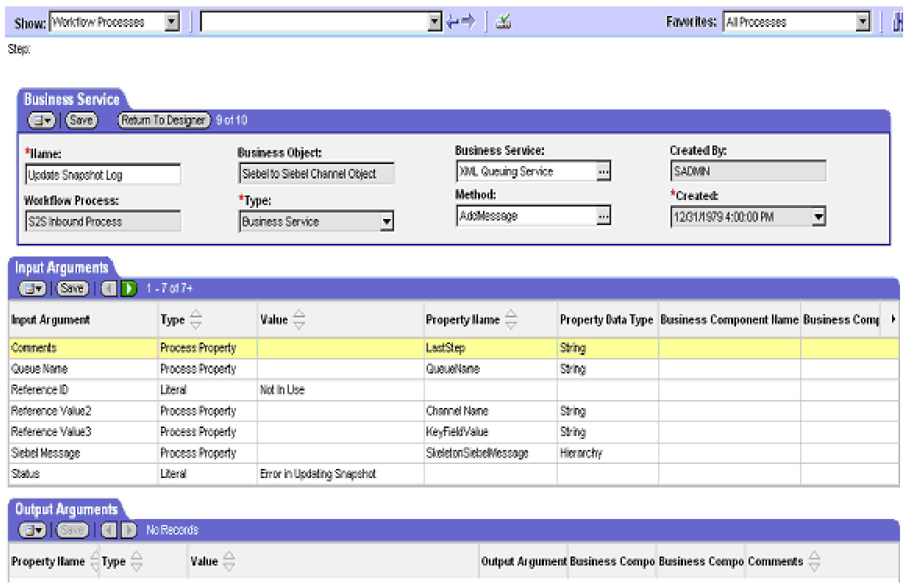
13 **Set Status:** This step invokes the Update Status method of Snapshot Access Service. This step is only visited when the Siebel Adapter fails to process the snapshot data. It sets a flag which indicates to the receiving process that this record needs to do error recovery.



14 **Log Failure:** All fatal errors are logged with status ErredProcess and the current workflow process is terminated.



15 **Update Snapshot Log:** If Step5: Update Snapshot fails and the error is nonfatal, workflow should continue. However, this fact is logged for the Siebel Administrator with status set to Processed.



Administration and Error Messages 6

This chapter describes the procedures to administer Siebel Connector for Siebel eBusiness Applications. Table 3 lists the tasks for administering Siebel Connector. Each of these tasks is covered separately in this chapter.

Table 3. Tasks for Administering Siebel Connector

Task	Where performed	When performed
Error Recovery and Administration on page 98	srvrmgr command-line	When there is an instance failure with hardware or software.
Adding New Instances on page 101	Siebel Connector	When another instance in the organization wants to join the Siebel Connector deployment.
Resolving Data Conflicts on page 102	Siebel Connector	As needed.
Performance on page 103	Siebel Connector	As needed.

Error Recovery and Administration

Generally, the Siebel Connector relies upon guaranteed delivery by the transport mechanism (MSMQ or MQ Series) to protect against data loss. If an instance failure occurs, use the following procedures given below to recover.

Recovering from a Hardware Failure

The steps presented here explain the process for recovering from a hardware failure.

To recover from a hardware failure

1. Restart the publishing workflow.
2. Restart the receiving workflow.

This should be sufficient for the application to pick up from the point where the failure occurred.

Recovering from a Software Failure

On the receiving side, a software failure is more likely than a hardware failure. Recovering from a software failure (more likely than a hardware failure) typically involves identifying the error, reporting the error to the publishing side, and correcting the data and republishing. These procedures are summarized below.

Handling Failure Errors

To identify the software failure

1. Navigate to Integration Administration screen.
2. Navigate to EAI Queues.

   The EAI Queues applet and EAI Queue Item applet appear.

3. Search for S2S Inbound Process Log Queue.

   The EAI Queue Item applet lists detailed information about the failure.

Table 4 explains the EAI Queue Item applet columns.

Table 4. EAI Queue Fields

Field	Description
Comments	Last step that errored out in the S2S Inbound Process
Queue Name	This is always S2S Inbound Process Log Queue
Reference ID	Normal or Delete Root
Reference Value2	Channel Name
Reference Value3	S-S Key ID value
Siebel Message	Siebel message that caused the error
Status	Error in Processing

If the EAI Queue Item applet show records, the error occurred as data was being received. In this case, examine the Reference Value2 and Reference Value3 columns for more information about the failed record.

To report the error to the publishing side

1. Review appropriate log files to identify the reason for the failure.
2. Send the error record channel name and S-S Key Id value to the publishing side administrator.

To correct the data and republish

1. Navigate to the Siebel to Siebel Connector > Global Data.
2. Locate the corresponding snapshot.
3. Set the Error Recovery flag to Y.

Handling Root Deletion Errors

In cases where deleting a root is attempted and failed, use the following procedure to recover.

To identify the error

1. Perform steps 1-3, under "To identify the software failure" on page 98.

2. When the EAI Queue Item applet is displayed, identify any records with Delete Root in the Reference Id column.

3. Run workflow to do error recovery:

   - From srvrmgr command line type the following command:

     ```
     start task for comp WfProcMgr with ProcessName='S2S Error
     Recovery Delete Root Process'
     ```

   The failed record is deleted from the database. The log is also deleted from the EAI Queue Item.

Adding New Instances

For instructions on how to create and add new instances to a Siebel eBusiness Application deployment, refer to procedures in Chapter 5, “Initializing a New Instance.”

Resolving Data Conflicts

This section describes the type of data error conflicts you may encounter and the procedure to resolve them. The following situations will generate data errors:

- Identification fields for global data, S-S Key Id, are different on two Siebel instances.
- User key fields are identical on two Siebel instances.

The following procedure describes how to resolve these conflicts.

To resolve data error conflicts

- Delete the record on the receiving instance.

Performance

This section presents techniques to help improve the performance of the Siebel Connector for Siebel eBusiness Applications.

Pagination

Siebel Connector for Siebel eBusiness Applications ships with the following pagination workflows. These workflows require less memory than their non-pagination counterparts.

- `S2S Outbound Process MQ - Pagination` for use with the IBM MQSeries transport.
- `S2S Outbound Process MSMQ - Pagination` for use with the MSMQ transport.

These workflows require less memory than their non-pagination counterparts because they process queries in smaller batches (100 integration object instances is the default) and because they serialize the sending process.

In contrast, non-pagination workflows query *all* object instances in a single batch and then send the result in a single, large message.

You should consider using a pagination workflow if:

- Memory resources are scarce

or

- A large number of integration object instances will be queried

You use these workflows in the same way as non-pagination workflows.

NOTE: You can customize the batch size by changing the default value for process property *PageSize* in the pagination workflow.

Multiple Channels

If you are publishing and receiving through a single channel and this does not meet your performance requirements, consider using multiple channels.

However, use caution when setting up multiple channels between two instances. Specifically, be sure you follow the recommendations explained below.

- Use *different* search specifications among all channels for the same integration object.

- Use search specifications that divide all data into disjoint sets, so that a given record is published through *one and only one channe*l.

  In other words, define searchspec1 and searchspec2, for example, in such a way that the query result for searchspec1 and the query result for searchspec2 will not contain the same record. This rule applies to one-to-many, many-to-one, and many-to-many relationships.

- Define search specifications in such a way that any record you want to send or receive meets *only one search specification at all times*.

  In other words, the record must not jump channels.

- Siebel recommends basing your search specifications on some invariant field—such as the S-S Key Id field—which, once created, never change value. Thus, if the record is modified in the future, it will not cause the record to jump channels because it now meets a different, and unintended, search specification.

Index