SIEBEL 7
eBusiness

SIEBEL FINANCIAL SERVICES CONNECTOR FOR IFX XML GUIDE

VERSION 7.0, REV. H

12-BD361Z

JULY 2002

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404
Copyright © 2002 Siebel Systems, Inc.
All rights reserved.
Printed in the United States of America

No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior agreement and written permission of Siebel Systems, Inc.

The full text search capabilities of Siebel eBusiness Applications include technology used under license from Fulcrum Technologies, Inc. and are the copyright of Fulcrum Technologies, Inc. and/or its licensors.

Siebel, the Siebel logo, TrickleSync, TSQ, Universal Agent, and other Siebel product names referenced herein are trademarks of Siebel Systems, Inc., and may be registered in certain jurisdictions.

Supportsoft™ is a registered trademark of Supportsoft, Inc. Other product names, designations, logos, and symbols may be trademarks or registered trademarks of their respective owners.

U.S. GOVERNMENT RESTRICTED RIGHTS. Programs, Ancillary Programs and Documentation, delivered subject to the Department of Defense Federal Acquisition Regulation Supplement, are "commercial computer software" as set forth in DFARS 227.7202, Commercial Computer Software and Commercial Computer Software Documentation, and as such, any use, duplication and disclosure of the Programs, Ancillary Programs and Documentation shall be subject to the restrictions contained in the applicable Siebel license agreement. All other use, duplication and disclosure of the Programs, Ancillary Programs and Documentation by the U.S. Government shall be subject to the applicable Siebel license agreement and the restrictions contained in subsection (c) of FAR 52.227-19, Commercial Computer Software - Restricted Rights (June 1987), or FAR 52.227-14, Rights in Data—General, including Alternate III (June 1987), as applicable. Contractor/licensor is Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404.

Proprietary Information
Siebel Systems, Inc. considers information included in this documentation and in Siebel eBusiness Applications Online Help to be Confidential Information. Your access to and use of this Confidential Information are subject to the terms and conditions of: (1) the applicable Siebel Systems software license agreement, which has been executed and with which you agree to comply; and (2) the proprietary and restricted rights notices included in this documentation.

Contents

Chapter 3. Configuration Roadmap

Appendix A. Additional Information

Introduction

This book will be useful primarily to people whose titles or job descriptions match one of the following:

Business Analysts	Persons responsible for analyzing application integration challenges and planning integration solutions at an enterprise.
Siebel Financial Services Application Administrators	Persons responsible for planning, setting up, and maintaining Siebel Financial Services applications.
Siebel Financial Services Application Developers	Persons responsible for planning, implementing, and configuring Siebel Financial Services applications, and possibly adding new functionality.
Siebel Financial Services 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 Financial Services 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.

Also included in the audience for this book is any user with experience in data integration, data transformation (data mapping), scripting or programming, and XML.

How This Guide Is Organized

This guide describes the Siebel Connector for IFX XML architecture and explains how to configure and use the Siebel Connector for IFX XML to integrate, share, and replicate data between Siebel Financial Services applications and external applications. The documentation for Siebel Financial Services eBusiness Application Integration and all the pre-built connectors includes:

- *Siebel Financial Services eBusiness Application Integration Guide*
- *Siebel Financial Services Connector for ACORD P&C and Surety Guide*
- *Siebel Financial Services Connector for IFX XML Guide*
- *Siebel Enterprise Integration Manager Administration Guide Addendum for Financial Services*

Additional Documentation

The product documentation set for Siebel Financial Services applications is provided on the *Siebel Financial Services Bookshelf* CD-ROM. For general information about Siebel product documentation, see *Siebel Financial Services Bookshelf*.

EAI-Associated Books. Review *Overview: Siebel eBusiness Application Integration Volume I* and *Siebel Financial Services eBusiness Application Integration Guide* to learn what Siebel eAI is offering as a base product for Siebel Financial Services eAI. Refer to *Siebel Object Interfaces Reference* if you plan on using COM, CORBA, or the ActiveX Plug-ins to accomplish integration. *Siebel Object Types Reference* serves as a reference for Siebel business objects and components. Read both *Siebel Tools Guide* and *Siebel Workflow Administration Guide*to gain an understanding of the Siebel tools required for creating integration. Read *Siebel Enterprise Integration Manager Administration Guide* and *Siebel Enterprise Integration Manager Administration Guide Addendum for Financial Services* if you will perform bulk loading or unloading of data. The Connector books provide specifics on each of the associated connectors.

Siebel Systems, Inc., reserves the right to modify the documentation for Siebel eBusiness Applications at any time. For updates to Siebel documentation, go to the SupportWeb site (http://ebusiness.siebel.com/supportweb/).

If you want to order additional Siebel documentation and copies of the *Siebel Financial Services Bookshelf* CD-ROM, go to Books Online at http://ebusiness.siebel.com/booksonline.

To access both SupportWeb and Books Online, you will need to provide the user name and password you received from Siebel Support Services (support@siebel.com).

Using the Siebel Product

It is strongly recommended that you read *Fundamentals* so that you can make optimal use of your Siebel application, especially if you are new to Siebel software. *Fundamentals* provides detailed coverage of the Siebel user interface and how to use it; working with data; locating information with the query and find features; sharing information with other users; and so on. The features presented in *Fundamentals* appear throughout the Siebel application suite; they are introduced through procedures you can learn and use in your own Siebel application.

Revision History

Siebel Financial Services Connector for IFX XML Guide, Version 7.0, Rev. H

1 Overview

The Siebel Connector for IFX XML provides integration between Siebel eBusiness Applications and other systems. The connector makes use of the IFX Business Specification document Version 1.0.1, a financial industry standard XML specification.

The Siebel Connector for IFX XML receives, parses, and processes the business operations specified in the XML message. It handles both outbound and inbound messages.

This integration offers powerful capabilities designed to support communication between a financial institution and its customers, its service providers, and other financial institutions. Eventually it will support direct communication between a financial institution and its customers. This solution allows you to effectively harness the synergies between Siebel front office applications and IFX-based applications. The Siebel Connector for IFX XML extends Siebel applications to integrate with back office data and business processes.

The Siebel Connector for IFX XML supports both synchronous and asynchronous transactions across application boundaries. The resulting consistency and sharing of data enables efficient coordination between front and back office operations. For example, a customer can request a balance inquiry in Siebel Financial Services eBusiness applications and receive real-time response from an IFX-enabled banking application within the bank. Similarly, an individual account holder may bank from home using a secure connection over the public Internet to connect to the financial institution's Siebel eBusiness application. The request message, sent from the customer's personal computer at home, may enter the financial institution's private data network for processing or be routed to a third-party service provider that processes that message on behalf of the financial institution. Regardless of the organization that actually processes the message or what computing and network architecture that organization has installed, the customer receives a response message with standard semantics.

Required Components

The Siebel Connector for IFX XML requires the following components in order to implement message exchanges between Siebel eBusiness Applications and IFX-compliant applications:

- Siebel Financial Services
- A license to use the Siebel Connector for IFX

  The Siebel Connector for IFX XML license key can be obtained from Siebel Manufacturing Operations. Please log a service request on Siebel SupportWeb.

  NOTE: Siebel Connector for IFX XML is not automatically available as part of Siebel Financial Services, but must be purchased separately.

- Siebel Event Manager to initiate a workflow process through a Siebel workflow manager (optional). In the absence of the event manager, an eScript can initiate a workflow process. Siebel Workflow is delivered as a part of Siebel Financial Services.

NOTE: You should also be familiar with IFX XML models. Additional information about these models can be obtained by visiting http://www.ifxforum.org.

For the purposes of this document, Siebel assumes that all these products have been successfully installed and tested for completeness by trained personnel before starting to use the Siebel Connector for IFX XML for integration. Please refer to Chapter 3, “Configuration Roadmap,” for implementing integration.

Using the Siebel Connector for IFX XML

This chapter provides a brief overview of the capabilities of the Siebel Connector for IFX XML. Additional information about integration with Siebel eBusiness Applications is available in:

- *Overview: Siebel eBusiness Application Integration Volume I*
- *Siebel Financial Services eBusiness Application Integration Guide*

Your work with the Siebel Connector for IFX XML consists of:

- Using the IFX wizard to create integration objects to map data between Siebel and IFX-based external applications.
- Creating integration workflows based on the mapped objects.

You can learn how to build the transformation maps and create workflows from this guide. You can also use some out-of-the-box IFX messages and workflows defined in this guide as your reference for implementation. Some information on customizing your integration is included in this guide, but you will also need to consult additional guides specified in the text.

Major chapters in this guide provide a description of IFX rules and syntax, the methods and arguments for configuring a Siebel Connector for IFX XML to customize your integration solution, and a sample implementation showing the steps involved to configure and use the connector.

Siebel Connector for IFX XML Architecture

The Siebel Connector for IFX XML is a configurable set of components that allow data to be exchanged between your Siebel application and external IFX-based applications and databases. As shown in Figure 1 on page 1-16, the Siebel Connector for IFX XML is built on top of the Siebel Financial Services eAI Architecture, which in turn is built on top of Siebel eBusiness Application Integration (eAI) Architecture. The Siebel Financial Services eAI framework has been built to support an XML messaging-based communication infrastructure.

Generally speaking, users of Siebel Financial Services must integrate with many different applications through messaging mechanisms. In order to fulfill this requirement, many connectors have to be built in order to support various industry standards. Siebel Financial Services is in a position to quickly and easily build and deploy multiple connectors based on the flexible Siebel eAI Architecture.

To demonstrate such flexibility, Siebel Systems has built two connectors in this release—ACORD P&C Connector and Siebel Connector for IFX XML—both based on the Siebel Financial Services EAI framework. Please refer to *Siebel Financial Services eBusiness Application Integration Guide* for more information about the flexible Siebel Financial Services architecture.

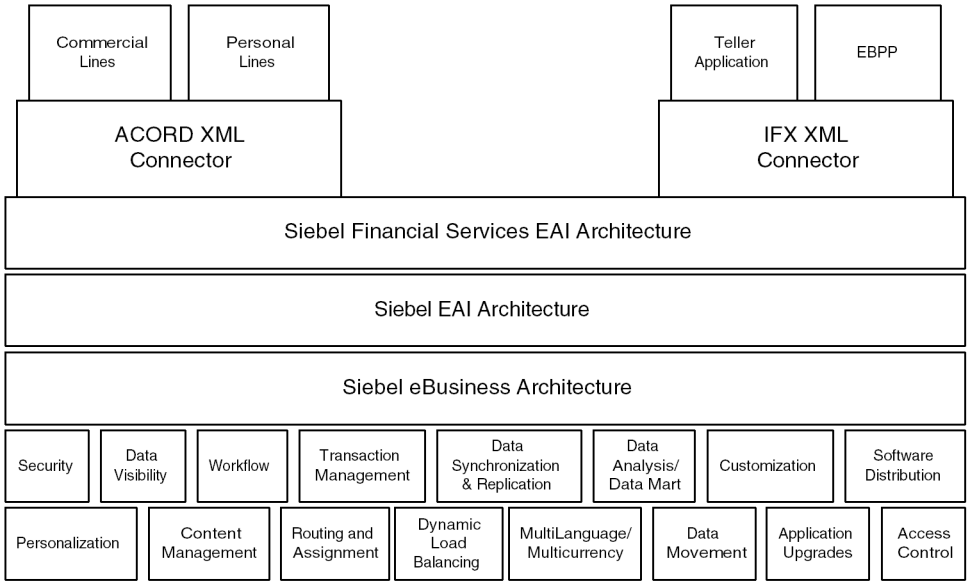


Figure 1. High-Level Architecture of Siebel Financial Services eAI

Financial Services architecture provides the functions required for XML processing, such as:

- Handling the XML message header
- Handling heterogeneous commands in the body section of an XML message
- Data type formatting and conversions
- Data model mapping through the various connector modules

These Siebel Connector for IFX XML modules include the FINS IFX Wizard, the FINS IFX XML Dispatcher, the FINS IFX XML Converter, the FINS IFX XML DTE, and the FINS IFX XML Transaction Manager.

The Siebel Connector for IFX XML is based on the IFX XML standard for financial industry data exchange. Siebel Connector for IFX XML was built using the IFX XML standard. The IFX XML standard is designed to address financial institutions' real-time requirement by defining banking transactions that include both a request and a response message. The IFX specification with this connector allows financial institutions to support customers using a broad range of channels, including, but not limited to:

- World Wide Web access using any standard web browser software
- Personal computers with personal financial manager (PFM) software
- Voice response units (VRUs) that provide bank by phone services
- Automated teller machines (ATMs)
- Consumer handheld devices such as personal digital assistants (PDAs)
- Mobile telephones with data capabilities

The Siebel Connector for IFX XML supports all data types in the IFX specification that are used to represent all data passed between clients and servers using the messages defined.

Business Data Flows

Each standard integration or custom integration is based on business data flows. A business data flow controls the transformation of an IFX-based data object to a Siebel data object and a Siebel data object to an IFX-based data object.

There are two types of business data flows:

- Outbound to an external IFX-based application (Send)
- Inbound from an external IFX-based application (Receive)

Figure 2 illustrates both inbound and outbound business data flows.

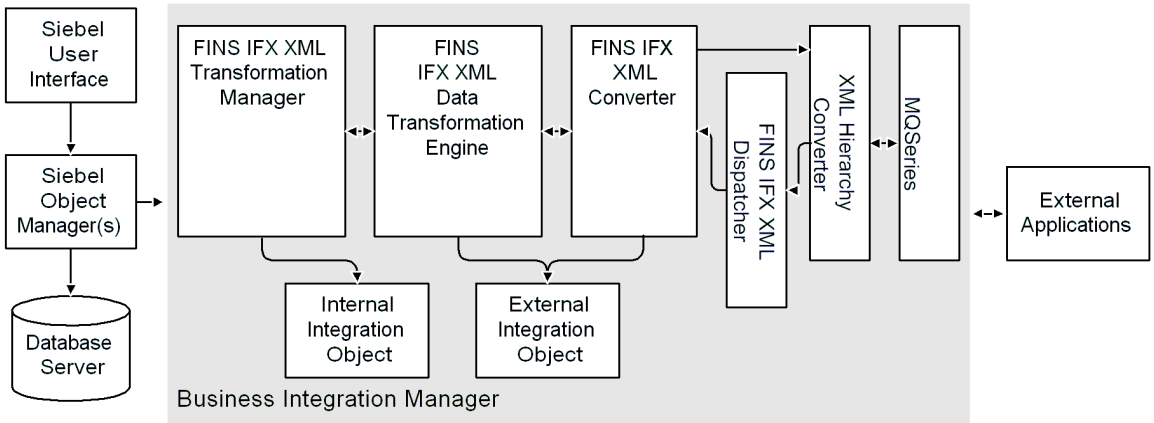


Figure 2. Data Flow in the Siebel Connector for IFX XML

The processing for each type of data flow is contained within a Siebel workflow. The workflow process is initiated by the Siebel Event Manager or by a Siebel eScript call.

Outbound Data Flow

Figure 2 on page 1-18 illustrates an outbound data flow as well as an inbound data flow. During an outbound data flow:

1. When the workflow is initiated, the FINS IFX XML Transaction Manager extracts data from the Siebel database. It takes as input all the ROW_IDs of the objects.
2. This data is then used to instantiate the internal integration objects based on the Siebel business objects.
3. The FINS IFX XML Transaction Manager then returns all the instances retrieved as Siebel property sets.

   A property set is a representation of data in memory in the Siebel internal format. It is used widely by the business services that constitute the connector components.
4. The internal integration object instances are then passed to the FINS IFX XML Data Transformation Engine (DTE) to transform the internal integration object instances into external integration object instances.

   The FINS IFX XML DTE also adds all necessary IFX-specific command layer attributes into the instances transformed.
5. The FINS IFX XML Converter converts all external integration object instances into proper XML integration object instances. It also adds the envelope, header, and other sections to the newly converted instance.
6. Lastly, the XML Hierarchy Converter converts the XML integration object instance from a property set format into a text format.
7. The message is then sent to external systems using any transport mechanism supported by Siebel eAI.

NOTE: Figure 2 on page 1-18 depicts the connector process using the MQ Series transport adapter. However, the transport mechanism can be HTTP, MSMQ, or any other transport mechanisms supported by Siebel eAI.

Inbound Data Flow

Figure 2 on page 1-18 illustrates an inbound data flow as well as an outbound data flow.

Inbound business data flows start with a receiver server component such as the MQSeries, HTTP, or MSQM.

The receiver runs in the background continuously, waiting for data from external IFX-based applications. When the receiver receives an IFX message, it invokes the workflow process configured to handle and process the data. The workflow typically dictates the whole Siebel Connector for IFX XML business logic.

1. The raw XML text string is passed through the XML Hierarchy Converter to be converted into an XML integration object instance.
2. The FINS IFX XML Dispatcher then takes in the XML instance, parses it and identifies the messages received according to the rule sets in the IFX Dispatcher Map. The IFX dispatcher identifies the envelope, header and body sections. The dispatcher then associates the appropriate internal and external integration objects to the message so that it can be processed by the converter.

   The Dispatcher map is an integration object created by the FINS IFX wizard.
3. The FINS IFX Converter then takes the XML instance, and processes individual sections of the instance while converting each sub-tree into external integration object instances.
4. The FINS IFX XML DTE transforms the external integration object instances into internal integration object instances.
5. The internal integration object instances are passed to the FINS IFX XML Transaction Manager which performs the operation specified in the instance through the invocation of other business services configured in its user properties.

Workflow Integration

Siebel workflows control the flow and transformation of data into and out of Siebel applications. You create a workflow using the Workflow Designer, a graphical user interface provided within the Siebel applications. Siebel workflows provide many more capabilities than those described in this guide. For more information about Siebel Workflow, see *Siebel Workflow Administration Guide*.

Integration Objects

Integration objects are the data containers used within the workflow environment. They represent the data structure of either a Siebel business object or an external application's data object.

You can create integration objects with the integration object wizard provided in Siebel Tools. The integration object wizard can create Siebel integration objects from Siebel business objects.

For IFX integration work, please use the FINS IFX Wizard in Siebel Tools that reads an IFX Document Type Definition (DTD) and creates the required external integration objects, pairs them with the internal integration objects, creates the envelope and header integration objects, and finally associates all of these in the rule-based dispatcher map.

This document describes how to use the FINS IFX Wizard to complete design time requirements. For more information about the FINS IFX Wizard see Chapter 2, "Siebel Connector for IFX XML," in this document. For more information on the integration objects read *Overview: Siebel eBusiness Application Integration Volume I.*

Business Services

All of the connector components are Siebel business services. Business services execute predefined or customized actions in a workflow process. Examples of business services include the FINS IFX XML Transaction Manager, Siebel eAI Adapter, and the FINS IFX Converter.

Siebel business services act on property sets passed to them. They perform business logic operations such as interfacing with a database, interfacing to IFX-based systems, or transforming one integration object into another.

Business services have object-like qualities, such as methods, method arguments, and user properties. These elements define how a business service can be used. Although business services can be used to perform many different functions, they all have a standard interface.

Siebel Systems, Inc., provides many business services, and you can create your own. Business services are defined in Siebel Tools.

This guide describes those business services used to interface to IFX-based systems. For more information on business services in general, read *Integration Platform Technologies: Siebel eBusiness Application Integration Volume II.*

IFX XML Standard

When handling insurance application information, your Siebel application implements the IFX standard for XML documents to connect with external applications.

IFX publishes a DTD that allows the Siebel Connector for IFX XML to create mappings between its data and data in external databases. The DTD is required by Siebel Connector for IFX XMLs.

You can find the IFX DTD, along with complete documentation, at the following location: http://www.ifxforum.com. Be certain to use the appropriate version of the IFX DTD, version 1.0.1.

Siebel Connector for IFX XML 2

This section describes the methods, input arguments, and output arguments for configuring the components of a Siebel Connector for IFX XML.

The Siebel Connector for IFX XML consists of the following components:

- Transaction Manager
- Transformation Engine
- Converter
- Dispatcher
- Transport Adapter

Figure 3 shows the connector components.

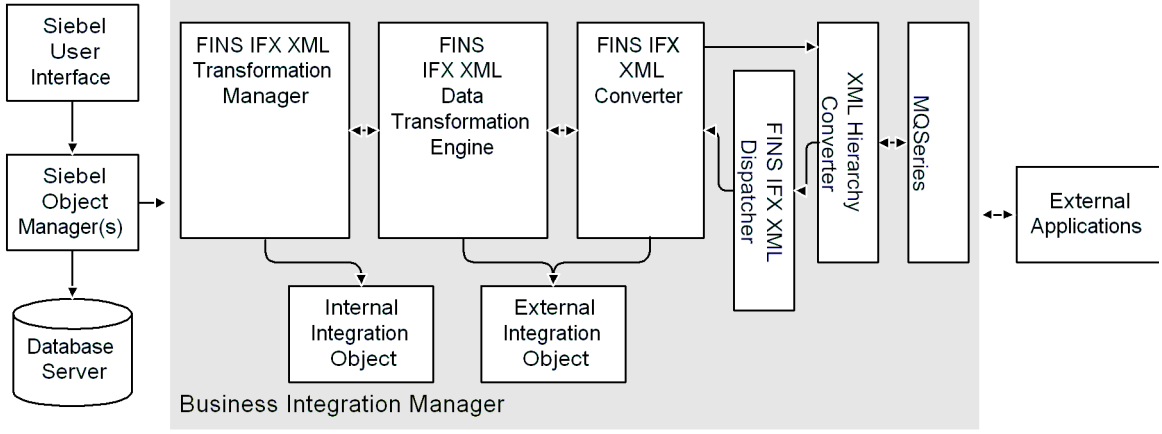


Figure 3. Siebel Connector for IFX XML Components

The Connector components are Siebel business services, which are configured in the Workflow view. The integration objects are created using the FINS IFX XML wizard, and they are configured using the Data Map editor.

NOTE: For information about Siebel integration objects, converter elements, and XML, see *XML Integration Reference*.

IFX XML Syntax and Rules

IFX is a financial-industry version of XML. It contains messages that are appropriate for the financial and banking industries.

The IFX standard defines the required structure and format of an XML message for use with a Siebel connector. The definition is in the IFX DTD, and the IFX DTD is incorporated by the Siebel connector to construct messages.

This section provides a summary of the IFX XML syntax and rules, and provides the appropriate vocabulary for discussing IFX XML messages. This section supplies knowledge that is basic for any troubleshooting you may need to do.

IFX XML Documents

Each IFX XML document has three distinct parts:

- Envelope
- Header
- Body

The parts are presented as a hierarchy: the envelope is the root, which contains the header and the body. Elements of an IFX XML document that contain other elements are called aggregates.

The envelope and header provide information required by the XML converter and by other components in the connector. The services identify the kind of business service affected by the information, and the messages provide the data that is being exchanged. There are elements that precede the message proper, which specify the versions of XML and IFX.

Figure 4 shows a sample IFX XML document:

Figure 4. Sample IFX XML Document

Envelope

The envelope is the root element of an XML document. For an IFX XML document, it begins with <IFX> and ends with </IFX>.

The indicator <IFX> is the only item in the envelope.

Header

Every message header has a sign-on element that authenticates the message, and it may have a sign-off element that ends a particular session.

The header has four elements:

- SignonRq
- SignonRs
- SignoffRq
- SignoffRs

The header for a request has the header element <SignonRq>. The header for the response has the header element <SignonRs>. Similarly, the sign-off elements are specifically for requests and responses.

NOTE: IFX XML messages must be either requests or responses. Requests and responses cannot be mixed in a single message. A request uses <SignonRq>. A response uses <SignonRs>.

Signon Information

The <SignonRq> or <SignonRs> header element provides a location for status information, authentication information, date and time stamps, language preferences, and identification of the application that will use the data. You can find complete information in the IFX specification.

Authentication Information

The *initial* <SignonRq> for any session must provide authentication information, typically the user name and password, or a certificate ID. When the server authenticates the user, using the information in the header, the server issues a session key in the <SignonRs>. Subsequent messages use the session key as a token. After a session has finished, any subsequent session must start with the authentication information again.

Following is an example of an initial SignonRs authentication element.

```
<SignonRq>
   <SignonPswd>
      <CustId>
         <SPName>com.siebel</SPName>
         <CustLoginId>RLIU</CustLoginId>
      </CustId>
      <CustPswd>
         <CryptType>NONE</CryptType>
         <Pswd>DROWSSAP</Pswd>
      </CustPswd>
      <GenSessKey>1</GenSessKey>
   </SignonPswd>
   <ClientDt>2001-11-16T16:56:39.0-08:00</ClientDt>
   <CustLangPref>ENU</CustLangPref>
   <ClientApp>
      <Org>Siebel FINS</Org>
      <Name>Siebel FINS Application</Name>
      <Version>7.0</Version>
   </ClientApp>
</SignonRq>
```

Additional elements may be included in a Signon element. You can find complete information in the IFX specification.

Status information, which includes error codes, may also appear in the Signon element. Status information is discussed in "Status Information and Error Codes" on page 2-34.

Signoff Information

The Signoff header element is used to end a session. A typical time to end a session is at the close of business for the day.

The Signoff element, <SignoffRq> or <SignoffRs>, appears at the end of the message, just before the end of the envelope </IFX>. The Signoff element may optionally contain a <custID> element.

Body

The body of an IFX XML document provides the content of the information request or response. The body serves as an aggregate containing services and messages. Services and messages, in turn, are aggregates that contain smaller elements.

- **Service.** A service identifies the kind of service being requested or delivered, identifying the business function that will be affected. For example, <PaySvcRq> is a request for a payment service, and <BankSvcRq> is a request for a bank service.
- **Message.** A message identifies the business object affected by the message and the operation that is to be performed on the data. For example, <PmtAddRq> is a request to add a payment.
- **Data Element.** A data element identifies the business component or fields affected by an operation defined in the message. For example, <FirstName> is a data element that contains information about a person's first name.

Services

The basic body element is a service, for example <PaySvcRq>, <BaseSvcRq>, or <BankSvcRq>. <BaseSvcRq> is a request for the base service, which all service providers can provide.

An IFX body can include multiple services. A body almost always contains at least one service. A body with no service would provide only authentication.

The same service may be included in a body more than once, but each service must be for a different service provider.

Following is an example of a message with a single payment service request.

```
<PaySvcRq>
   <SPName>Partner IFX Middleware</SPName>
   <RqUID>50DBF4F7-7888-480b-927E-333652FEBF87</RqUID>
   <PmtAddRq>
      <RqUID>BD620AC4-53E7-4UIL-588C-YOR8D6224FE9</RqUID>
      <PmtInfo>
         <RemitInfo>
            <CustPayeeId>0VF-VEBQ</CustPayeeId>
            <CurAmt>
               <Amt>2500</Amt>
               <CurCode>USD</CurCode>
            </CurAmt>
         </RemitInfo>
         <CardAcctIdFrom>
            <AcctId>2574-86392</AcctId>
            <AcctType>Savings/MMA</AcctType>
         </CardAcctIdFrom>
         <DueDt>2001-11-13</DueDt>
      </PmtInfo>
      <DupChkOverride>1</DupChkOverride>
   </PmtAddRq>
</PaySvcRq>
```

The service aggregate includes a universally unique identifier (UUID) to match responses to requests. The UUID is generated using an algorithm that makes it unique. It appears in the <RqUID> element. It is generated by the client (which sends out the request). It is stored at the client site, which then matches it to the UUID in the response message. The UUID generator can be a Siebel business service or an extension provided by a third party. In any case, the UUID generator is identified by a parameter to the IFX Converter.

Messages

Messages (sometimes called business messages) are contained in Service aggregates. Each service can contain one or more business messages. Each service can contain any number of messages.

The message tag identifies the business object that is affected by the message and a command operator. A business object can be a payment or a cash drawer—anything on which an operation can be performed.

A message uses one of the following operations:

- Add
- Delete
- Cancel
- Inquiry
- Modify
- Audit
- Synchronize

The business message name tag contains the object and the operation. For example, a business message called <PmtAddRq> identifies "payment" as the business object, and "add" as the operation. The details of the added payment are provided within the message.

A complete list of business messages for IFX XML is provided in the IFX XML implementation specification.

Data Elements

Within the business message are additional elements that identify the record that should be affected by the request or response and provide any other specifications, such as <CustName>, <PostAddr>, <FirstName>, and <LastName>.

The additional elements include field labels, field information, and tags that provide program access to the data.

Following is an example of data elements for the add payment request.

```
<PmtAddRq>
   <RqUID>BD620AC4-53E7-4UIL-588C-YOR8D6224FE9</RqUID>
   <PmtInfo>
      <RemitInfo>
         <CustPayeeId>0VF-VEBQ</CustPayeeId>
         <CurAmt>
            <Amt>2500</Amt>
            <CurCode>USD</CurCode>
         </CurAmt>
      </RemitInfo>
      <CardAcctIdFrom>
         <AcctId>2574-86392</AcctId>
         <AcctType>Savings/MMA</AcctType>
      </CardAcctIdFrom>
      <DueDt>2001-11-13</DueDt>
   </PmtInfo>
   <DupChkOverride>1</DupChkOverride>
</PmtAddRq>
```

The information in this request is sent to the external application, which performs the request and returns a response.

Status Information and Error Codes

Status information is information about the current status of a message. It appears only in response documents. It can appear in a response header or in any element of a response body.

The external server inserts status information after processing the document. If the processing is satisfactory, status information may or may not be inserted. If there is a problem in the processing, the status information identifies the problem.

A status code of zero means the status is satisfactory. If any other number appears, it is an error code or warning, and the message is flagged. The error code can be used in troubleshooting.

Status information in the header applies to the entire IFX XML document. Status information in a service applies to that service. Status information in a message applies to that message. The following example shows status information in a header.

```
<Status>
   <StatusCode>100</StatusCode>
   <Severity>Error</Severity>
   <StatusDesc>General Error</StatusDesc>
</Status>
```

For details of status codes, see the IFX XML specification, which provides a description of all error codes.

FINS IFX XML Wizard

Siebel eBusiness applications provide wizards to guide you through the process of building integration objects and updating dispatcher maps.

You can use the FINS IFX XML Wizard to build integration objects for the Siebel Connector for IFX XML. The wizard guides you through the process of selecting objects (from the Siebel repository or from an external system) on which you can base your new Siebel integration object. The wizard builds a list of valid components from which you choose the specific components to be included in your Siebel integration object.

You access Siebel wizards within the Siebel Integration Object Builder in Siebel Tools. Use the IFX XML wizard to create an appropriate elements hierarchy that reflects the IFX XML DTD. The wizard:

- Creates a set of integration objects to handle outbound and inbound messages and to handle internal and external integration.
- Updates the dispatcher map, which is later used by the dispatcher.

Integration Objects

Siebel integration objects allow you to represent integration metadata between a Siebel business object and an external XML standard, using the IFX XML DTD. The integration object represents a common structure that the eAI infrastructure can understand.

Because these integration objects adhere to a set of structural conventions, they can be traversed and transformed programmatically, using Siebel eScript objects, methods, and functions, or transformed declaratively using Siebel data mapper.

To use Siebel Connector for IFX XML to integrate data you need to build three different integration objects:

- **IFX XML Envelope Integration Object.** An envelope integration object provides envelope and header information for an IFX XML document.

  User properties in an IFX XML envelope provide flexibility to the connector. For example, when a user sends an initial IFX XML request, the IFX XML document uses a <SignonRq> header that is different from subsequent <SignonRq> headers.

- **IFX XML Internal Integration Object.** An internal integration object represents the Siebel business object hierarchy for a particular Siebel business object. See Chapter 3, "Configuration Roadmap," for an example of creating an internal integration object.

- **IFX XML External Integration Object.** An external integration object represents the IFX XML hierarchy for a particular IFX XML message. See Chapter 3, "Configuration Roadmap," for an example of creating an external integration object.

FINS IFX XML Dispatcher Map

The dispatcher map is used by FINS IFX XML Dispatcher. The dispatcher map is another integration object that provides a rule set for handling incoming IFX XML messages. The dispatcher map is created and updated by the FINS IFX XML Wizard during the process of creating an external integration object.

The map contains information that associates message instances with the appropriate internal and external integration objects for incoming and outgoing messages. It associates each incoming or outgoing message with all the Siebel Connector for IFX XML elements that are necessary to translate it into Siebel data.

The map contains DTE map names, the internal integration object name, the external integration object name, and Siebel adaptor operations. These elements make up the translation scheme for the message instance. The dispatcher map allows the dispatcher to associate the proper translation scheme with each message instance.

All the mapping information is stored in the user property part of the dispatcher map integration object.

IFX XML Transaction Manager

The FINS IFX XML Transaction Manager is responsible for retrieving data from a Siebel application. It may invoke the Siebel adapter or another business service configured in its user properties. It is an adapter that resides logically between the Siebel object manager and the rest of the connector. It executes operations specified in an XML message instance as Siebel database transactions.

The transaction manager translates XML command elements into Siebel eAI Adapter actions. The transaction manager either carries out the action or finds another business service to carry out the action.

The transaction manager combines return results into a single property set. A property set is an intermediate data store that can be used in subsequent operations within the connector.

For inbound processing, the transaction manager accepts an IFX XML property set, which may contain multiple integration object instances for multiple transactions. It pairs each individual transaction request with an integration object instance and invokes methods in the Siebel eAI Adapter.

For outbound processing, the transaction manager pairs a transaction request with an integration object instance and sends an IFX XML property set to the DTE.

Transaction Manager User Properties

Table 1 describes the FINS IFX XML Transaction Manager user properties.

Table 1. FINS IFX XML Transaction Manager User Properties

Name	Value	Description
DispatcherMapName	< Integration object name >	The dispatch map name. Transaction manager will use this map to tag the Body information for other component. This value can be set as runtime input argument, which will overwrite this value.
< Operation Name >	< Service name > / < Method name/ < Argument List >	< OperationName > can be any literal value you want to use to name the operation. The operation can be invoked from the MethodName method in ServiceName business service passing Argument arguments. < OperationName > is an alias for the method specified by ServiceName/ MethodName. < OperationName > is referenced in dispatcher map entries. For information about configuring the < Operation Name > , see the following section.

Configuring the <Operation Name> Property

Use a meaningful name for the operation name, such as “IXMLOperation_Query.”

The value must follow this format:

- "Service/Method/Argument;Argument;"
- "/Method/Argument;Argument;"
- Note that the service, method, and argument are separated by a slash (/)
- Each argument ends with semi-colon (;)
- The default service name is "EAI Siebel Adapter"
- The default argument name is "SiebelMessage"

Uses of the Siebel operation include the following:

- "EAI Siebel Adapter/Query/PrimaryRowId;!SiebelMessage;SearchSpec;"
- "EAI Siebel Adapter/Query/#XMLHierarchy;"
- "EAI Siebel Adapter/Delete/RollbackOnSame;"

Table 2. Operation Examples

Example	Meaning
!SiebelMessage	The default value is to use SiebelMessage as the type of integration object instance. !SiebelMessage means to not use the default value.
#XMLHierarchy	Replace SiebelMessage with XMLHierarchy

Transaction Manager Methods and Arguments

The FINS IFX XML Transaction Manager methods and arguments are described in the following tables. Table 3 describes the FINS IFX XML Transaction Manager methods.

Table 3. FINS IFX XML Transaction Manager Methods

Method	Display Name	Function
Execute	Execute Transaction	Can be used for inbound or outbound messages when the integration object instance is provided. When only Row_Id is available, use the Execute Outbound method.
Execute Outbound	Execute Outbound	Use only for executing an outbound message.

Table 4 describes the arguments for the Execute Outbound method.

Table 4. Method Arguments for Execute Outbound

Argument	Value	Description
DispatcherMapName	< Integration object name >	Required input string. The name of the dispatcher map that contains the target IFX message.
IsVBC	true, false	Optional input string. Value is TRUE if the source Business Component is a VBC.
IXMLMapPath	< IFX absolute path >	Stores the key for looking up a dispatcher map entry. Transaction Manager uses it to look up the entry value for the integration object instance. Absolute path of the target IFX message.
PrimaryRowId	< row_id >	The primary row ID of the integration object.
SearchSpec	< Search spec >	The search specification of a query for the operation business service to retrieve an integration object instance from the Siebel database.
SiebelFINSOperationOut	< Operation name >	The operation to be used by the transaction manger, which is predefined in the user properties of the transaction manager.

Table 4. Method Arguments for Execute Outbound

Argument	Value	Description
VBCFieldMap	< Property Set >	Optional input hierarchy. A hierarchy contains the field values for the target VBC.
XMLHierarchy	< XML property set >	A property set that contains an IFX message instance in Siebel internal integration object format.

Table 5 provides specifications for the Execute Outbound method arguments.

Table 5. Argument Specifications for Execute Outbound Method

Name	Display Name	Data Type	Type	Optional
IXMLMapPath	IXML Map Path	String	Input	No
PrimaryRowId	Primary Row Id	String	Input	No
SiebelFINSOperationOut	Outbound operation	String	Input	No
SearchSpec	Search Spec	String	Input	Yes
XMLHierarchy	XML Property Set	Hierarchy	Output	No

Table 6 describes the arguments for the Execute method.

Table 6. Method Arguments for Execute Method

Argument	Value	Description
OnlyIOI	true, false	For an inbound message, the integration object instance for request may contain header, body, and envelope portions. When the transaction manager takes the proper operation against the Siebel application, the integration object instance for response is generated as well. If this value is set to TRUE, all information from the request message is dropped; in this case, the converter and DTE do not need to deal with the information overhead. If this value is set to FALSE, request information is carried over.
XMLHierarchy	< XML property set >	A property set that contains an IFX document instance in Siebel internal integration object format.

Table 7 provides specifications for the Execute method arguments.

Table 7. Argument Specifications for Execute Method

Name	Display Name	Data Type	Type	Optional
Only IOI	Produce only an integration object instance	String	Input	No
XMLHierarchy	XML Property Set	Hierarchy	Input or Output	No

FINS IFX XML Data Transformation Engine (DTE)

The FINS IFX XML DTE transforms property sets in a Siebel internal integration hierarchy to an external integration object hierarchy, and vice versa. This function allows the FINS IFX XML Converter to exchange data between two systems with different data models. The transformation map is defined at run time from Siebel Administration views.

For inbound processing, the DTE accepts a property set from the FINS IFX XML Converter and transforms it into a property set to be used by the FINS IFX Transaction Manager. The incoming property set is made up of one or more external integration object instances. If there are multiple instances, the DTE parses them into individual instances and transforms them. The DTE then packages the returned transformed instances as an output property set as internal integration object instances.

For outbound processing, the DTE accepts a property set from the transaction manager and transforms it into a property set to be used by the converter. The outgoing property set is made up of one or more internal integration object instances. The DTE then packages the returned transformed instances as an output property set as external integration object instances.

DTE Methods and Arguments

The FINS IFX XML DTE methods and arguments are described in the following tables. Table 8 describes the methods for the FINS IFX XML DTE.

Table 8. FINS IFX XML DTE Methods

Method	Display Name	Function
ToExternal	Transform to External Hierarchy	Transforms a Siebel hierarchy into an external hierarchy.
ToInternal	Transform to Siebel Hierarchy	Transforms an external hierarchy into a Siebel hierarchy.

Table 9 describes the arguments for the ToExternal method.

Table 9. Method Arguments for the ToExternal Method

Argument	Value	Description
XMLHierarchy	< XML property set >	Takes as input the output of the Execute outbound method of the IFX XML transaction manager. Output hierarchy that contains the IFX Document in Siebel external integration object format.
< MapArgs >		Runtime input arguments that can be used by DTE maps. See explanation in the following section.
DTE Argument 1	< Any literal value >	
DTE Argument 2	< Any literal value >	
DTE Argument 3	< Any literal value >	

Table 10 provides specifications for the ToExternal method arguments.

Table 10. Argument Specifications for ToExternal Method

Name	Display Name	Data Type	Type	Optional
XMLHierarchy	XML Property Set	Hierarchy	Input Output	No
< MapArgs >		String	Input	Yes
DTE Argument *n*				Yes

Table 11 describes the arguments for the ToInternal method.

Table 11. Method Arguments for the ToInternal Method

Argument	Value	Description
XMLHierarchy	< XML property set >	Takes as input the output of the XMLPropetySetToPropertySet method of the IFX Converter. Output hierarchy that contains the IFX Document in Siebel internal integration object format.
< MapArgs >		Runtime input arguments that can be used by DTE maps. See explanation in the following section.
DTE Argument 1	< Any literal value >	
DTE Argument 2	< Any literal value >	
DTE Argument 3	< Any literal value >	

Table 12 provides specifications for the ToInternal method arguments.

Table 12. Argument Specifications for ToInternal Method

Name	Display Name	Data Type	Type	Optional
XMLHierarchy	XML Property Set	Hierarchy	Input Output	No
< MapArgs >		String	Input	Yes
DTE Argument *n*				

Using <MapArgs>

< MapArgs > is a runtime input argument used by the DTE map to match an integration map argument of an integration object map. The FINS IFX XML DTE can take as many < MapArgs > as needed as long as each name is unique among all the < MapArgs > that are passed to the FINS IFX XML DTE at the same time.

For example, suppose that the output integration object instance has some fields mapping to a workflow process property, such as an ID field.

1. Using the Data Map view, select the integration map to edit in the Integration Object Map applet.
2. In the Integration Map Argument applet, create the map and set the following values:
   - Name = CompId
   - Data Type = “DTYPE_TEXT”
   - Display Name = Component ID
3. In the Integration Field Map applet, set the following values:
   - Target Field Name = [Id]
   - Source Expression = [&CompId]
4. In the workflow, set the data transformation engine input argument as follows:
   - Input Argument = CompId
   - Type = Process Property
   - Property Name = Object Id

At runtime, the DTE replaces [&CompId] with the value of the Object ID.

For some mappings, if the DTE cannot find the source field value, the DTE creates empty tags by default. To remove the empty tags, add IgnoreEmptyTag as the map argument.

For complete information, see *Business Processes and Rules: Siebel eBusiness Application Integration Volume IV*.

FINS IFX XML Converter

The purpose of the FINS IFX XML Converter is to generate and process IFX XML-specific elements, such as the <SignonRq> aggregate and the <SignonRs> aggregate.

The FINS IFX XML Converter receives hierarchy output and converts it into a property set or an XML string.

Converter User Properties

Table 13 describes the FINS IFX XML Converter user properties.

Table 13. Converter User Properties

Name	Value	Description
PI_Parameter: <PI_Name>	<PI_Value>	PI_Parameter: is a constant prefix. PI_Name is provided by the configurator. "PI_Name = PI_Value" would be a PI name-value pair included in IFX PI. Zero or more pairs can be defined. Examples: ■ PI_Parameter:newfileuid ■ PI_Parameter:oldfileuid ■ PI_Parameter:version
PI_Type	ifx	Process Instruction Type
XMLEnvIntObjectName	<Integration object name>	Integration object name that defines the IFX envelope.
ExceptionForIFXErr	true, false	If true, IFX converter generates an exception when any <StatusCode> in the incoming response message contains a non-zero value. This behavior can be overridden by providing a ProcessStatus outcall function. If a ProcessStatus outcall is provided, this flag does not take effect whether its value is true or false.

Converter Methods and Arguments

The FINS IFX XML Converter methods and arguments are described in the following tables. Table 14 describes the FINS IFX XML Converter methods.

Table 14. FINS IFX XML Converter Methods

Method	Display Name	Function
PropSetToXML	PropSetToXML	Generate the XML message to be sent.
PropSetToXMLPropSet	PropSetToXMLPropSet	Prepare the DOM structure of the XML message to be sent.
XMLPropSetToPropSet	XMLPropSetToPropSet	Convert the XML message received into hierarchical property sets.
XMLToPropSet	XMLToPropSet	Prepare the hierarchical property sets from DOM structure of the XML message received.
Generate Envelope Only	Generate Envelope Only	Generate an IFX document that contains only an IFX envelope and header. No body portion.

Table 15 describes the arguments common to FINS IFX XML Converter methods.

Table 15. Method Arguments for the FINS IFX XML Converter

Argument	Values	Description
Client Application Name	< Client application name >	Required input string. Value of < Name > of < ClientApp > aggregate.
Client Application Organization	< Client organization name >	Required input string. Value of < Org > of < ClientApp > aggregate.
Client Application Version	< Client application version >	Required input string. Value of < Version > of < ClientApp > aggregate.

Table 15. Method Arguments for the FINS IFX XML Converter

Argument	Values	Description
Date Input Format	< Date format >	Optional input string. Default value: YYYY-MM-DD
Date Output Format	< Date format >	Optional input string. Default value: YYYY-MM-DD
DateTime Input Format	< Date, time format >	Optional input string. Default value: YYYY-MM-DDTHH:mm:ss.0Z
DateTime Output Format	< Date, time format >	Optional input string. Default value: YYYY-MM-DDTHH:mm:ss.0Z
Time Input Format	< Time format >	Optional input string. Default value: HH:mm:ss.0Z
Time Output Format	< Time format >	Optional input string. Default value: HH:mm:ss.0Z
YrMon Input Format	< Year, month format >	Optional input string. Default value: YYYY-MM
YrMon Output Format	< Year, month format >	Optional input string. Default value: YYYY-MM
Enable Data Formatting	true, false	Optional input string. Default value: false. See additional explanation in Appendix A, "Additional Information."

Table 15. Method Arguments for the FINS IFX XML Converter

Argument	Values	Description
IFX Application Business Service Name	< Business service name >	Optional input string. Business service name that contains out-called functionality. If no value is entered, the connector uses default functionality. See the next section for further information about outcalls.
Is Log Out	true, false	Optional input string. Set to “true” for signing off IFX session.
Is Client	true, false	Optional input string. Set to “true” if connector is used at client side. Default value: “True”.
Service Provider Name	< IFX service provider name >	Optional input string. Value of < SPName > of < xxxSvcRq > aggregate.
XMLEnvIntObjectName	< Integration object name >	Required input string. Name of the integration object that defines IFX envelope.
XML Property Set	< XML hierarchy >	Required input hierarchy.

Table 16 provides specifications for the PropSetToXML method arguments.

Table 16. Argument Specifications for the PropSetToXML Method

Name	Display Name	Date Type	Type	Optional
Client Application Name	Client Application Name	String	Input	No
Client Application Organization	Client Application Organization	String	Input	No

Table 16. Argument Specifications for the PropSetToXML Method

Name	Display Name	Date Type	Type	Optional
Client Application Version	Client Application Version	String	Input	No
Date Output Format	Date Output Format	String	Input	Yes
DateTime Output Format	DateTime Output Format	String	Input	Yes
Time Output Format	Time Output Format	String	Input	Yes
YrMon Output Format	YrMon Output Format	String	Input	Yes
Enable Data Formatting	Enable Data Formatting	String	Input	Yes
IFX Application Business Service Name	IFX Application Business Service Name	String	Input	Yes
Is Log Out	Is Log Out	String	Input	Yes
Is Client	Is Client	String	Input	Yes
Service Provider Name	Service Provider Name	String	Input	Yes
XMLEnvIntObjectName	XMLEnvIntObjectName	String	Input	Yes
XML Property Set	XML Property Set	String	Input	Yes
XML Property Set	XML Property Set	String	Output	No

Table 17 provides specifications for the PropSetToXMLPropSet method arguments.

Table 17. Argument Specifications for the PropSetToXMLPropSet Method

Name	Display Name	Date Type	Type	Optional
Client Application Name	Client Application Name	String	Input	No
Client Application Organization	Client Application Organization	String	Input	No
Client Application Version	Client Application Version	String	Input	No
Date Output Format	Date Output Format	String	Input	Yes
DateTime Output Format	DateTime Output Format	String	Input	Yes
Time Output Format	Time Output Format	String	Input	Yes

Table 17. Argument Specifications for the PropSetToXMLPropSet Method

Name	Display Name	Date Type	Type	Optional
YrMon Output Format	YrMon Output Format	String	Input	Yes
Enable Data Formatting	Enable Data Formatting	String	Input	Yes
IFX Application Business Service Name	IFX Application Business Service Name	String	Input	Yes
Is Log Out	Is Log Out	String	Input	Yes
Is Client	Is Client	String	Input	Yes
Service Provider Name	Service Provider Name	String	Input	Yes
XMLEnvIntObjectName	XMLEnvIntObjectName	String	Input	Yes
XML Property Set	XML Property Set	String	Input	Yes
XML Property Set	XML Property Set	String	Output	No

Table 18 provides specifications for the XMLPropSetToPropSet method arguments.

Table 18. Argument Specifications for the XMLPropSetToPropSet Method

Name	Display Name	Date Type	Type	Optional
Date Input Format	Date Input Format	String	Input	Yes
DateTime Input Format	DateTime Input Format	String	Input	Yes
Time Input Format	Time Input Format	String	Input	Yes
YrMon Input Format	YrMon Input Format	String	Input	Yes
Enable Data Formatting	Enable Data Formatting	String	Input	Yes
IFX Application Business Service Name	IFX Application Business Service Name	String	Input	Yes
XML Property Set	XML Property Set	String	Input	No
XML Property Set	XML Property Set	String	Output	No

Table 19 provides specifications for the XMLToPropSet method arguments.

Table 19. Argument Specifications for the XMLToPropSet Method

Name	Display Name	Date Type	Type	Optional
Date Input Format	Date Input Format	String	Input	Yes
DateTime Input Format	DateTime Input Format	String	Input	Yes
Time Input Format	Time Input Format	String	Input	Yes
YrMon Input Format	YrMon Input Format	String	Input	Yes
Enable Data Formatting	Enable Data Formatting	String	Input	Yes
IFX Application Business Service Name	IFX Application Business Service Name	String	Input	Yes
XML Property Set	XML Property Set	String	Input	Yes
XML Property Set	XML Property Set	String	Output	No

Table 20 provides specifications for the Generate Envelope Only method arguments.

Table 20. Argument Specifications for the Generate Envelope Only Method

Name	Display Name	Date Type	Type	Optional
Client Application Name	Client Application Name	String	Input	No
Client Application Organization	Client Application Organization	String	Input	No
Client Application Version	Client Application Version	String	Input	No
Date Output Format	Date Output Format	String	Input	Yes
DateTime Output Format	DateTime Output Format	String	Input	Yes
Time Output Format	Time Output Format	String	Input	Yes
YrMon Output Format	YrMon Output Format	String	Input	Yes
Enable Data Formatting	Enable Data Formatting	String	Input	Yes
IFX Application Business Service Name	IFX Application Business Service Name	String	Input	Yes

Table 20. Argument Specifications for the Generate Envelope Only Method

Name	Display Name	Date Type	Type	Optional
Is Log Out	Is Log Out	String	Input	Yes
Is Client	Is Client	String	Input	Yes
Service Provider Name	Service Provider Name	String	Input	Yes
XMLEnvIntObjectName	XMLEnvIntObjectName	String	Input	Yes
XML Property Set	XML Property Set	String	Input	Yes
XML Property Set	XML Property Set	String	Output	No

Outcalls

The Siebel Connector for IFX XML provides default behaviors for a variety of methods (shown in Table 21). If the default behaviors do not suit your needs, the Siebel Connector for IFX XML allows outcalls to override the defaults.

For example, the Siebel default for the <SignonRq> method is user name and password. The configurator can provide an outcall function for a certificate instead.

The outcall function is activated by the IFX Application Business Service Name parameter. If you want to use an outcall, you define a business service that encloses the outcall functionality, and then enter the business service name in the IFX Application Business Service Name parameter.

If the connector cannot find the business service identified in the IFX Application Business Service Name parameter, default functionality is used instead.

Siebel Connector for IFX XML Outcall Methods

This section lists all methods that the Siebel Connector for IFX XML can use for outcalls. You can implement them using either eScript or VB. For more details, see *Siebel eScript Language Reference* and *Siebel VB Language Reference*.

In Table 21, the input argument values are provided by the Siebel Connector for IFX XML, and the output argument values are returned to the Siebel Connector for IFX XML.

Table 21. Connector Outcall Methods

Method	Input Argument	Output Argument	Remarks
GenerateSignonRq	An empty property set	Property set contains < SignonRq > aggregate in property set format	Generate a customized < SignonRq > aggregate instead of having the IFX connector generate a default < SignonRq > aggregate.
GenerateSignonRs	Complete IFX request document in property set format	Property set contains < SignonRs > aggregate	Generate a customized < SignonRs > aggregate instead of having the IFX connector generate a default < SignonRs > aggregate.
GenerateSignoffRq	An empty property set	Property set contains < SignoffRq > aggregate in property set format	Generate a customized < SignoffRq > aggregate instead of having the IFX connector generate a default < SignoffRq > aggregate.
GenerateSignoffRs	Complete IFX request document in property set format.	Property set contains < SignoffRs > aggregate.	Generate a customized < SignoffRs > aggregate instead of having the IFX connector generate a default < SignoffRs > aggregate.
GenerateUUID	A property set with Type = GUID	A property set with generated GUID stored as its < Value >	Generate a valid GUID to be < RqUID > .
ProcessSignonRq	Complete IFX request document in property set format	Not required[1]	Process < SignonRq > .
ProcessSignonRs	Complete IFX request document in property set format	Not required	Process < SignonRs > .
ProcessSignoffRq	Complete IFX request document in property set format	Not required	Process < SignoffRq > .

Table 21. Connector Outcall Methods

Method	Input Argument	Output Argument	Remarks
ProcessSignoffRs	Complete IFX response document in property set format	Not required	Process < SignoffRs > .
ProcessStatus	An IFX response message	Not required	Process < Status > . This method is called once for every response message.
FormatFieldFromXML	A property set with ■ Type: IFX data type for this element ■ Value: Element value in IFX data format	A property set with ■ Value: Element value in Siebel data format	Convert element value from IFX data format to Siebel data format. This method is called for every element when data formatting is enabled.
FormatFieldToXML	A property set with ■ Type: IFX data type for this element ■ Value: Element value in Siebel data format	A property set with ■ Value: Element value in IFX data format	Convert element value from Siebel data format to IFX data format. This method is called for every element when data formatting is enabled.

1. Process methods do not require output values because the connector is not expecting any value in return. For example, if StatusCode is not 0, you may want to process the information in your own application, but no value is expected by the FINS IFX XML Converter.

FINS IFX XML Dispatcher

The FINS IFX XML dispatcher handles inbound XML hierarchy instances. It provides the necessary information for subsequent modules to perform their operations, such as the integration objects to be used.

The dispatcher identifies incoming messages and parses them into header and envelope sections. It also analyzes incoming message body sections, walking through each command. Using the dispatcher map, the dispatcher associates the message with the appropriate external integration object so that the FINS IFX XML Converter can use it. It also associates the message with the DTE map so that the FINS IFX XML DTE can use it.

Dispatcher User Properties

Table 22 shows the user properties for the dispatcher.

Table 22. User Properties for the FINS IFX XML Dispatcher

Name	Value	Comments
DispatcherMapName	< Integration object name >	Name of an integration object that details the dispatching rules and syntax for the IFX XML standard. This map is usually created along with all the other integration objects needed by the wizard.
XMLEnvIntObjectName	< Integration object name >	Name of an integration object that defines the content and hierarchy for the envelope and header sections of IFX XML.

Dispatcher Methods and Arguments

The FINS IFX XML Converter methods and arguments are described in the following tables. Table 23 describes the FINS IFX XML Dispatcher method.

Table 23. Dispatcher Method

Method	Display Name	Function
Dispatch Message	Dispatch Method	Validates the incoming XML message. if the message conforms to the dispatching rules, the integration object names and other necessary information will be attached to the message.

Table 24 describes the arguments for the Dispatch Message method.

Table 24. Method Arguments for the Dispatch Message Method

Argument	Value	Description
DispatcherMapName	< Integration object name >	Required input string
XMLEnvIntObjectName	< Integration object name >	Required input string
XML Hierarchy	< XML property set >	Required input hierarchy
XML Hierarchy	< XML property set >	Required output hierarchy

Table 25 provides specifications for the Dispatch Message method arguments.

Table 25. Argument Specifications for Dispatch Message Method

Name	Display Name	Data Type	Type	Optional
DispatcherMapName	DispatcherMapName	String	Input	No
XMLEnvIntObjectName	XMLEnvIntObjectName	String	Input	
XML Hierarchy	XML Hierarchy	Hierarchy	Input	
XML Hierarchy	XML Hierarchy	Hierarchy	Output	

Transport Adapter

The transport adapter is a Siebel business service that provides the interface between the outside data source and the Siebel connector. The connector can use any of the following standard transport mechanisms.

- MQSeries
- MQSeries AMI
- HTTP
- MSMQ

For details about the transport adapter, see *Transports and Interfaces: Siebel eBusiness Application Integration Volume III*.

Configuration Roadmap 3

This chapter provides an illustrative example of configuring the Siebel Connector for IFX XML.

The Siebel Connector for IFX XML is made up of four pre-built business services:

- FINS IFX XML Transaction Manager
- FINS IFX XML Data Transformation Engine
- FINS IFX XML Converter
- FINS IFX XML Dispatcher

The Siebel Connector for IFX XML can be configured to support several types of IFX Business Object Model packages. Figure 5 illustrates the main steps in configuring the Siebel Connector for IFX XML.

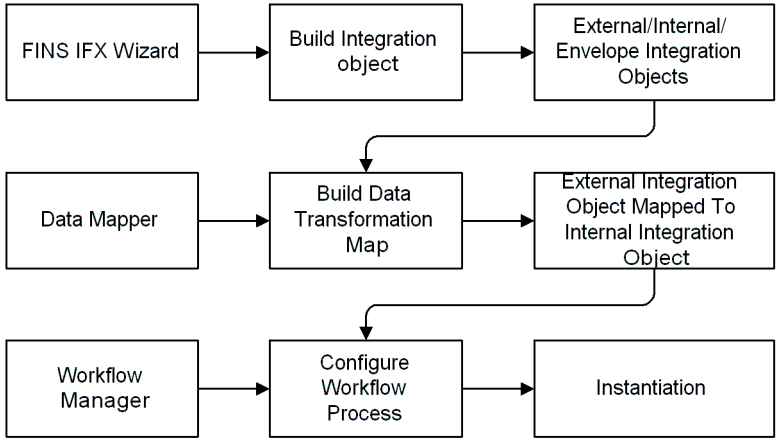


Figure 5. Main Steps to Configure the Siebel Connector for IFX XML

This chapter presents the scenario of adding a new Payment Schedule account service request through a Siebel front-end application for outbound communication. This operation corresponds to < PmtAddRq > and < PmtAddRs > messages in IFX XML. The example uses the HTTP Transport mechanism, though there is no specific transport mechanism required for Siebel Connector for IFX XML.

The following checklist shows the high-level procedure for configuring your system to use the Siebel Connector for IFX XML.

Checklist

q	Create a new project in Siebel Tools. *For details, see* "Preparing Project Elements" on page 3-65.
q	Instantiate the wizard for creating external and envelope objects for the new project in Siebel Tools. *For details, see* "Creating an IFX Wizard Business Service for the Project" on page 3-67.
q	Create integration objects in Siebel Tools. *For details, see* "Creating Integration Objects in Siebel Tools" on page 3-70.
q	Configure the IFX XML business services in Siebel Tools. *For details, see* "Configuring the Connector Components" on page 3-87.
q	Configure the transformation maps in Siebel Client. *For details, see* "Configuring the Data Transformation Maps" on page 3-92.
q	Configure an outbound Siebel connector in the Workflow Designer. *For details, see* "Configuring the Workflow Process" on page 3-104.
q	Configure runtime events to trigger the workflow process in real time according to user input using Runtime Event Manager. *For details, see* "Configuring Runtime Events" on page 3-124.

NOTE: When generating integration objects, be certain to use the appropriate version of the IFX DTD, version 1.0.1. It is available from www.ifxforum.com.

Preparing Project Elements

This section describes how to create the following:

- A dedicated project for the tutorials in this roadmap.
- A dispatcher map integration object for storing dispatcher map entries that are created by FINS IFX Wizard.
- A FINS IFX Tutorial Wizard business service for the tutorials in this chapter.

Creating a New Project

This tutorial starts by creating a new project to be used through for the remainder of the procedures. Creating a new project is optional. A locked project is essential. You may want to lock a project that already exists. The following tutorials assume that the same locked project is used throughout.

For complete information about creating a new project, see *Siebel Tools Reference*.

To create a project

1. Start Siebel Tools, if necessary.
2. Click the Project icon in Object Explorer.
3. Right click in the Projects window, and select New Record in the pop-up menu.
4. Fill in the following values for the new project record:
   - Name = FINS IFX Tutorial
   - Comments = FINS IFX Guide Tutorial
5. Click on the Locked field to lock this new project.

Creating an Empty Integration Object for Dispatcher Map

The locked project must contain integration object to be used as a dispatcher map. The dispatcher map contains the rule sets used by the FINS IFX XML Dispatcher.

Creating a new integration object is optional, but your project must contain a dispatcher map.

For complete information about creating a new integration object, see *Siebel Tools Reference*.

To create an integration map object

1. Start Siebel Tools, if necessary.
2. Click on the Integration Object icon in Object Explorer.
3. Right click in the Integration Objects window and select New Record in the pop-up menu.
4. Fill in the following values for the new integration object record:
   - Name = FINS IFX Tutorial DispMap
   - Project = FINS IFX Tutorial
   - Base Object Type = None
   - External Name = FINS IFX Tutorial DispMap
   - Comments = Dispatch map for entries created by FINS IFX Tutorial Wizard

Creating an IFX Wizard Business Service for the Project

We will be creating a new FINS IFX Tutorial Wizard business service based on the same class that FINS IFX Wizard Service is based on.

Creating a new FINS IFX wizard business service is optional.

You can customize the FINS XML integration object wizards to fit your needs, or you can create new ones.

In general, in the following situations you may decide to customize the available wizard:

- When you need to use a dispatcher map other than the one provided.
- When you need to use different Siebel operations for the same types of messages.

In general, in the following situations you may decide to create a new wizard:

- When you need to work in parallel and you want your integration objects to be created in different Tools projects. All integration objects are created in the same Tools project that the Wizard business service is in. To do this, copy the Wizard business service to the new project and rename it.
- When the same message needs to be sent from events that are based on different internal integration objects, which means you need the same message entry in different dispatcher maps. To do this, you could manually adjust the dispatcher map instead of creating a new Wizard business service.

Once you have customized the existing wizard or created a new wizard, you will compile the definition into the repository that Siebel Tools is using. Changes do not take effect prior to compiling.

Table 26 shows the pre-setup user properties for the FINS IFX Wizard.

Table 26. FINS IFX Wizard User Properties

Name	Value	Comments
DispatcherMapName	IFXDispMap	The dispatcher map name. The wizard will use this map to update the key and value.
HasUIControl	true	Internal use.
Integration Object Wizard	Y	Internal use.
Integration Object Base Object Type	Siebel Business Object	Internal use.
Operation KeyWord Match:0	Add/SAUpsert	Internal use. This means that when the wizard generates an external integration object for an Add message, it defines the operation in the transaction manager as SAUpsert. The operation name will be recorded in the dispatcher map.
Operation KeyWord Match:1	Inq/SAQuery	Internal use. This means that when the wizard generates an external integration object for an Inq message, it defines the operation in the transaction manager as SAUpsert. The operation name will be recorded in the dispatcher map.
Default Envelope Tag	IFX	Value for Envelope Tag.

NOTE: You can define a new Operation KeyWord Match:X if you need to. For example, if the IFX DTD in the future supports the an additional operation, you can define Operation KeyWord Match:2 as < new operation > /SA < New operation name > .

For complete information about creating a new business service, changing user properties, or compiling objects, see *Siebel Tools Reference*.

To create an IFX Wizard business service

1. Start Siebel Tools, if necessary.
2. Select FINS IFX from Project drop-down menu in Object Explorer.
3. Click the Business Service icon in Object Explorer.
4. Select FINS IFX Wizard Service in Business Services window.
5. Right-click FINS IFX Wizard Service record, and select Copy Record from the pop-up menu.
6. Fill in the following values for the new business service record:
   - Name = FINS IFX Tutorial Wizard Service
   - Project = FINS IFX Tutorial
   - Class = CSSFAIFXUIService
   - Display Name = FINS IFX Tutorial Wizard
7. Change the value of the DispatcherMapName user property for this new business service from IFXDispMap to FINS IFX Tutorial DispMap.
8. Validate and compile all the work you have done so far to the SRF file that your Siebel Tools uses.
9. Restart Siebel Tools.

Creating Integration Objects in Siebel Tools

Integration objects define the intermediate format of the data so that it can be used by the connector components to translate between Siebel data formats and IFX XML data formats.

You use the FINS IFX Wizard to create the internal, and external integration objects, as well as the dispatcher map.

Copying the Envelope Integration Object

An envelope integration object provides the envelope and header information needed by the IFX XML hierarchy.

The following procedure shows how to copy and modify the envelope integration object that works with external and internal integration objects. The FINS IFX Envelope v101 uses < SignonPswd > as the initial signon mechanism.

To copy and modify the envelope integration object

1. In Siebel Tools, choose View > Options.
2. Click the Object explorer tab, move Integration Object to the Visible Top Level Objects list, and click OK.
3. In the Type list, click the Integration Object icon.

   Integration objects are displayed on right hand side.
4. Query for FINS IFX Envelope v101.
5. When the record appears, right-click and choose Copy Record.
6. In the new record, fill in the following fields, if necessary:
   - Name = FINS IFX Tutorial Envelope
   - Project = FINS IFX Tutorial

   When the record is completed, go on to the next section to create external integration objects that will be used with this envelope integration object.

Creating External Integration Objects

An external integration object establishes the hierarchy for an IFX XML message. Each type of message under a service needs its own integration object that defines the body portion of the XML document.

When you create an external integration object, you create a pair of such objects, one for the request portion of the cycle and one for the response portion of the cycle.

Each external integration object is paired with an internal integration object when you configure the DTE map.

To create an external integration object

1. In Siebel Tools, lock the appropriate project, if necessary.

   The FINS IFX XML Wizard requires a locked project. This example uses the FINS IFX Tutorial project.

2. From the application-level menu, choose File > New Object.

   The New Object Wizards dialog box appears.

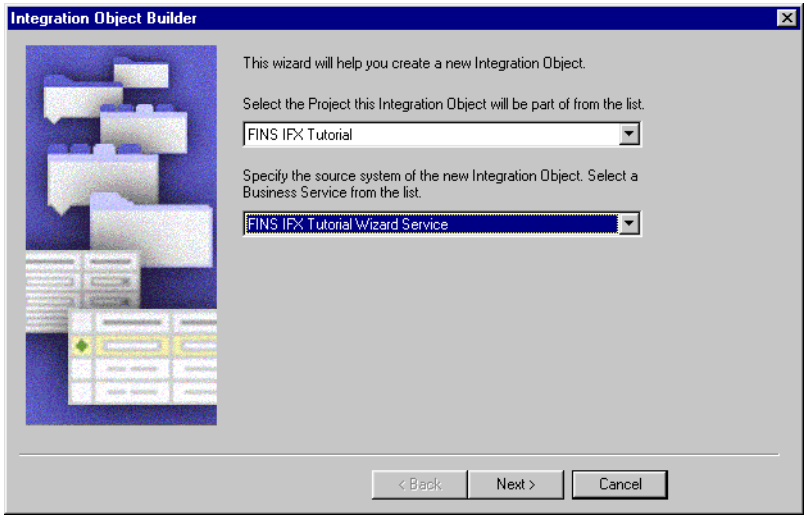
3. Select the EAI tab and double-click Integration Object.

   The Integration Object Builder wizard appears.

4 Fill in the two fields.

   a Select the project you have locked.

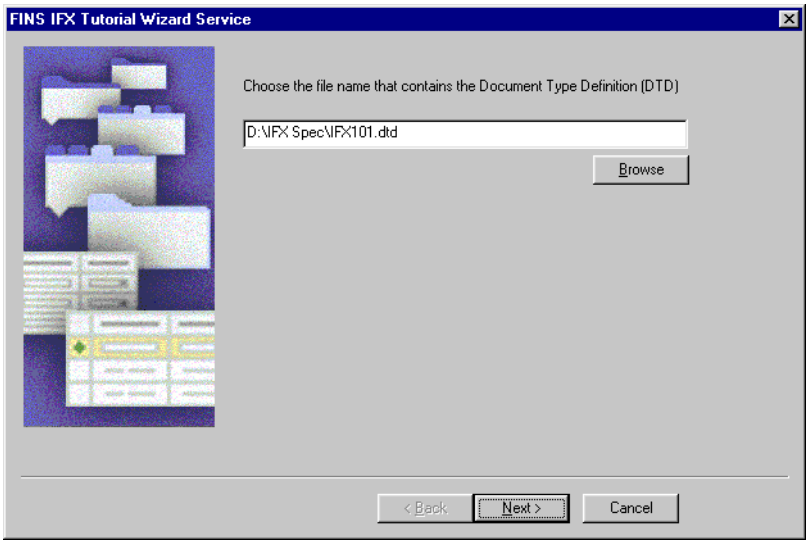
   b Select FINS IFX Tutorial Wizard Service from the Business Service list.



5 Click Next.

6 Choose the DTD file you want to use, and then click Next.

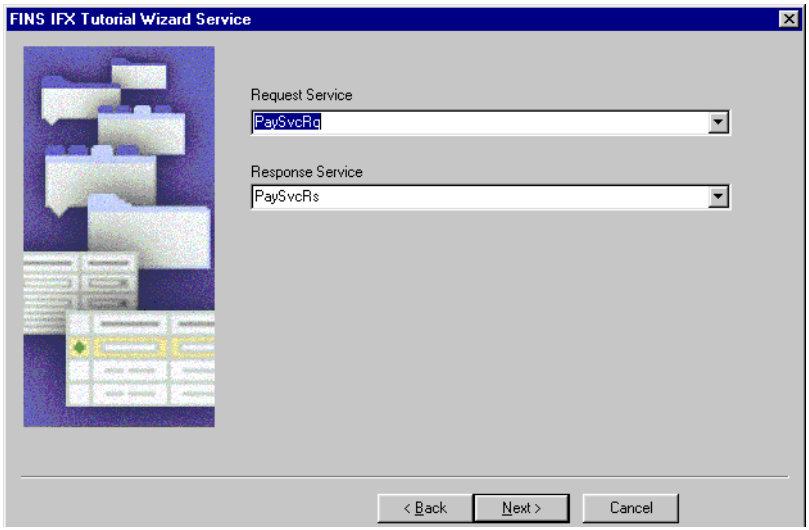
The filename is IFX101.dtd. This filename indicates that this is version 1.0.1 of IFX DTD.



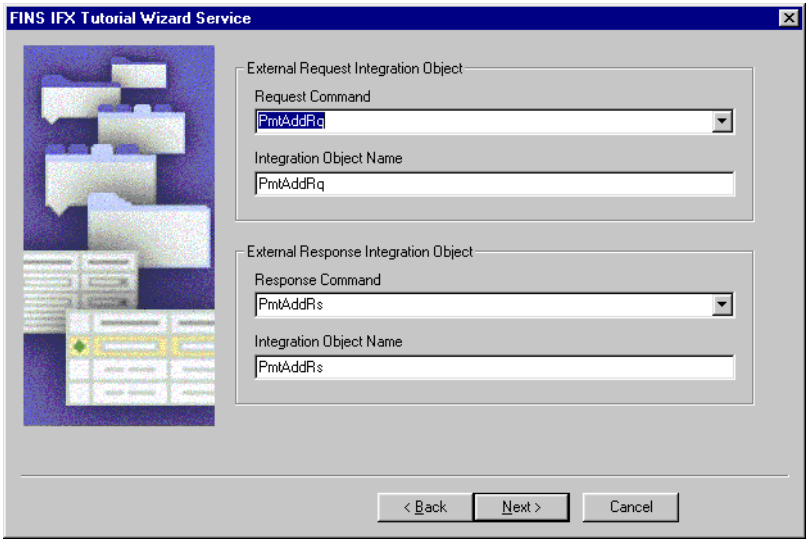
It takes some time for the wizard to parse the DTD file and to display the next page.

7 Choose a Request Service and Response Service pair.

Choose PaySvcRq for the Request Service. PaySvcRs is automatically entered for the Response Service.



8 Click Next, and then choose the Request Command and Response Command you want to use.



This screen uses Request Command to identify the IFX request message. The request message you select is automatically paired with an appropriate response message. For this example, you choose < PmtAddRq > ; it is automatically paired with < PmtAddRs > .

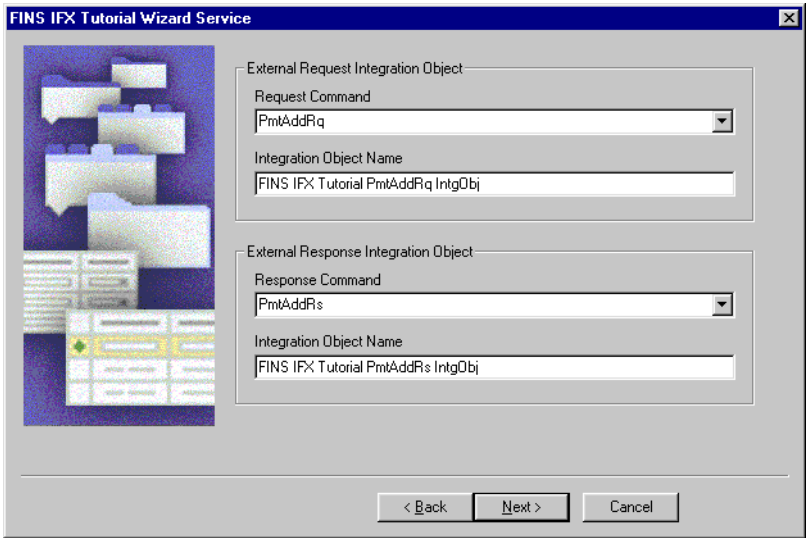
NOTE: Be sure to select the request message from the list. If the message is typed into the field, the wizard does not populate the remaining fields automatically.

You can change the Integration Object Name for the request and response integration objects for administrative convenience. You should consider establishing a set of naming conventions to make groups of objects easy to recognize. This example renames the external integration objects.

Integration object names for request and response messages must be unique within the repository. Note the integration object names. You will need to know the names when you configure the DTE map.

9 Change the integration object names for the request and response messages.

- Request object = FINS IFX Tutorial PmtAddRq IntgObj
- Response object = FINS IFX Tutorial PmtAddRs IntgObj

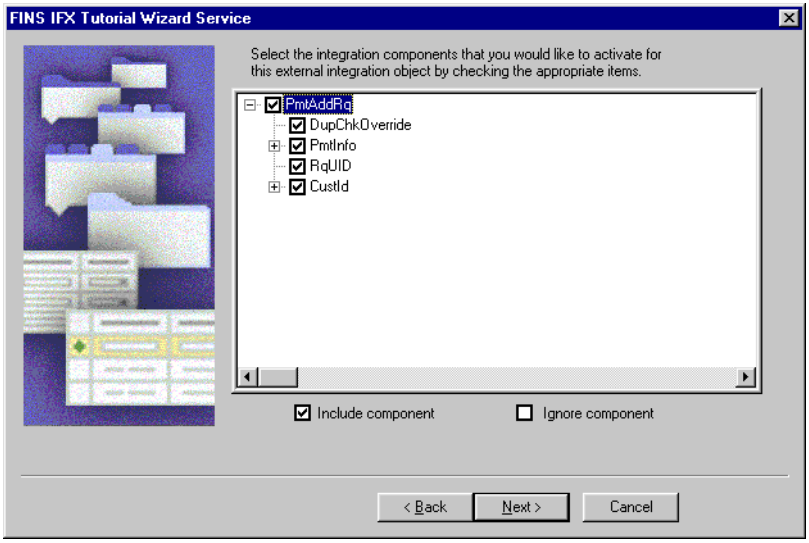


10 Click Next to display the integration components screen in which you select the message elements to include.

You will select message elements for the request integration object on this screen, and you will select the message elements for the response integration object in the next screen. The request integration object is FINS IFX Tutorial PmtAddRq IntgObj.

11 Click the plus symbol (+) to display the message elements.

This screen displays a visual hierarchy of the message structure. It provides all the available aggregates and elements for the message. The screen starts with all of these selected (included).

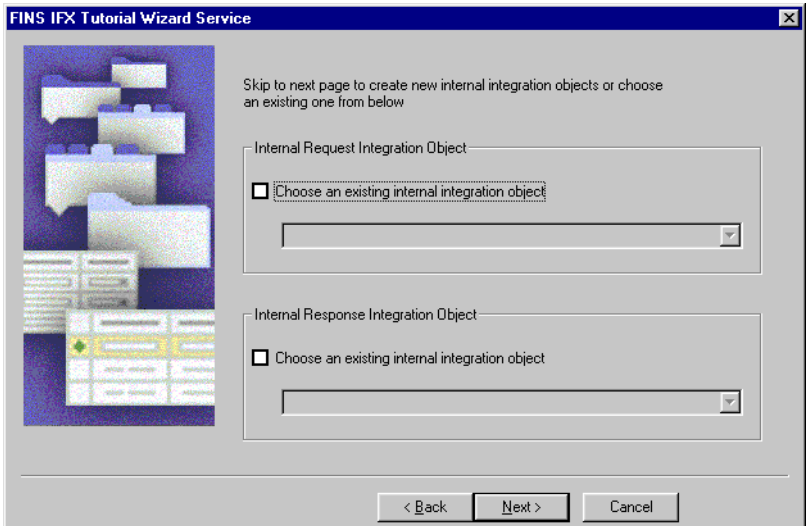


This example includes all the elements in the external integration objects. You would deselect elements to create an object to exchange fewer elements. Notice that if you deselect the parent, all the child items are deselected. Reselecting the parent does not reselect the child items, so you can select just a subset of child items.

12 Be sure that all the elements are selected, then click Next.

The second integration components screen allows you to select the message elements to include for your response message integration object. The response integration object is FINS IFX Tutorial PmtAddRs.

13 Click Next to include all elements. and display the Select Internal Integration Object screen.



See Appendix A, "Additional Information," for technical details about integration components, such as those created in this procedure.

Creating an Internal Integration Object

An internal integration object creates a structure that matches the data structure of a Siebel business object.

You can choose an already-created internal integration object, if one has been created. Siebel integration objects are interfaces for outside systems to interact with internal Siebel data.

This example uses the FINS IFX Wizard to create a new internal integration object, as shown in the following procedure. The new integration object will be paired with the external integration objects created in the previous section.

To create an internal integration object

1. Lock the project and select the DTD file for the object.

   You select the IFX101.dtd file.

2. Create a pair of external integration objects.

   The steps are described in the preceding sections.

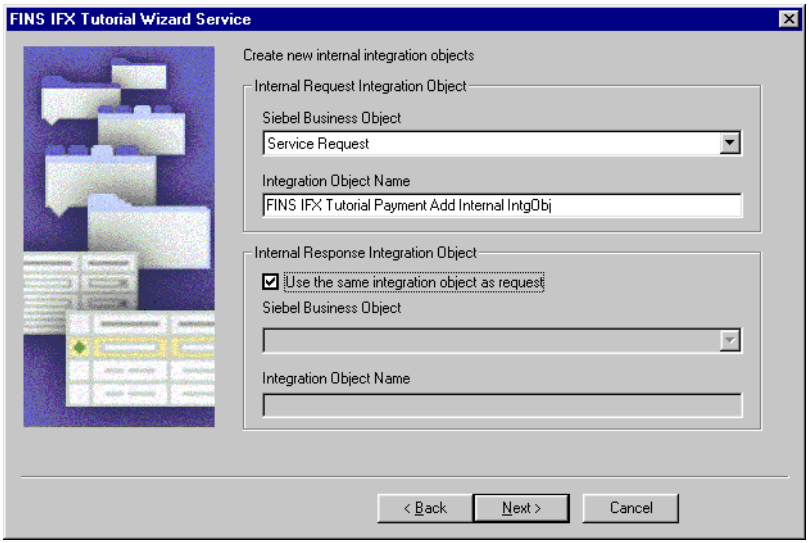
3. In the Select Internal Integration Object screen, do not select either check box, and click Next to display the New Integration Object screen.

4. In the Internal Request Integration Object area, choose the business object that contains the information that the connector will exchange, and enter the name of the integration object.

   For this example, choose the Service Request business object, which contains the information that the connector will exchange, and enter FINS IFX Tutorial Payment Add Internal IntgObj for the integration object name.

5 For the Internal Response Integration Object, click the Use the same integration object as request check box.

The entry boxes are grayed-out.



6 Click Next to display the integration components available from the business object you selected on the previous screen.

This screen displays a visual hierarchy of the business object structure. It provides all the available aggregates and elements for the message.

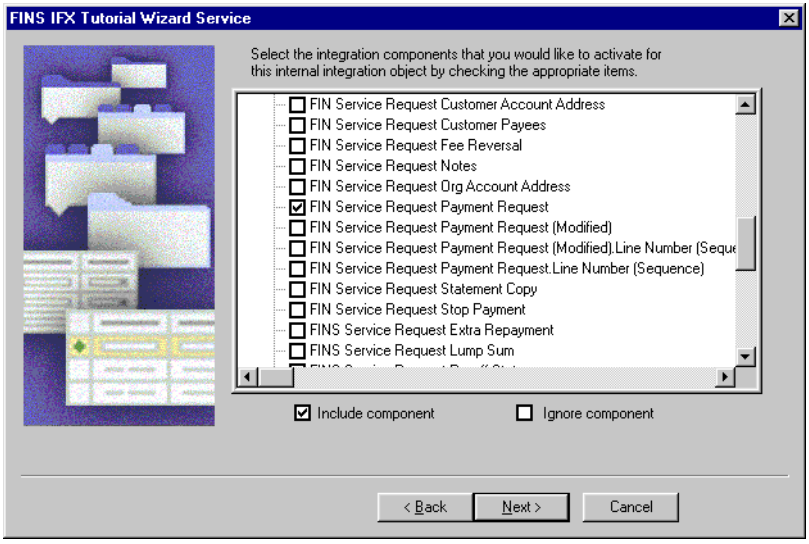
7 Click (+) to expand the list.

8 Deselect the root business component, Service Request.

Doing so deselects all business components in the hierarchy.

9 Select the FIN Service Request Payment Request business component.

The FIN Service Request Payment Request is the only business component needed for this tutorial.


FINS IFX Tutorial Wizard Service
Select the integration components that you would like to activate for this internal integration object by checking the appropriate items.
FIN Service Request Customer Account Address
FIN Service Request Customer Payees
FIN Service Request Fee Reversal
FIN Service Request Notes
FIN Service Request Org Account Address
FIN Service Request Payment Request
FIN Service Request Payment Request (Modified)
FIN Service Request Payment Request (Modified).Line Number (Seque
FIN Service Request Payment Request.Line Number (Sequence)
FIN Service Request Statement Copy
FIN Service Request Stop Payment
FINS Service Request Extra Repayment
FINS Service Request Lump Sum
Include component
Ignore component
< Back
Next >
Cancel

10 Click Next.

The screen displays a warning telling you that it may take some time to create the integration objects.

11 Click Yes to create the integration objects.

After the wizard creates the integration objects, it shows the objects that have been created.

Three new integration objects are generated in the FINS IFX Tutorial project and two dispatcher map entries are written into FINS IFX Tutorial DispMap integration object.

- Integration Objects:
  - FINS IFX Tutorial Payment Add Internal IntgObj
  - FINS IFX Tutorial PmtAddRq IntgObj
  - FINS IFX Tutorial PmtAddRs IntgObj
- Dispatcher Map entries:
  - IFX/PaySvcRq/PmtAddRq
  - IFX/PaySvcRs/PmtAddRs

Viewing the Dispatcher Map

When it creates the paired external and internal integration objects, the FINS IFX Tutorial Wizard creates a pair of new or updated entries in the dispatcher map.

The FINS IFX Tutorial DispMap dispatcher map is an integration object that contains the rule sets used by the FINS IFX XML Dispatcher. The user properties of the dispatcher map are used for storing dispatcher map entries.

In the following tutorial steps, the dispatcher map entries are explained, and then the entry for the response message is modified.

To view the dispatcher map user properties

1. From Siebel Tools, choose Object Explorer > Integration Object.
2. Query for the dispatcher map name, for example FINS IFX Tutorial DispMap.
3. Navigate to the user properties of the dispatcher map to see its user properties.

   To navigate to user properties, expand the sub-object types for the integration object and select User Properties.

   The following table shows the rule sets created by the wizard for the Add Payment scenario.

Name	Value
IFX/PaySvcRq/PmtAddRq	IFX/PaySvcRq/ PmtAddRq;PmtAddRq_ERqIRqMapIn;PmtAddRq_IRqERq MapOut;FINS IFX Tutorial PmtAddRq IntgObj;FINS IFX Tutorial Payment Add Internal IntgObj; IXMLOperation_ADD
IFX/PaySvcRs/PmtAddRs	IFX/PaySvcRs/ PmtAddRs;PmtAddRs_ERsIRsMapIn;PmtAddRs_IRqERq MapOut;FINS IFX Tutorial PmtAddRs IntgObj;FINS IFX Tutorial Payment Add Internal IntgObj; IXMLOperation_ADD

   The name of the user property represents the rule the dispatcher tries to match and the value represents the value the dispatcher needs to insert. For example, the name `IFX/PaySvcRq/PmtAddRq` is the path the dispatcher uses to locate the message received, and if it finds the match then it uses the information in the value column,

```
IFX/PaySvcRq/
PmtAddRq;PmtAddRq_ERqIRqMapIn;PmtAddRq_IRqERqMapOut;FINS IFX
Tutorial PmtAddRq IntgObj;FINS IFX Tutorial Payment Add
Internal IntgObj; IXMLOperation_ADD
```

to determine the action it needs to take. Following is a description of the meaning of each of the parts of the information in the value column.

Each value is made up of six tokens that are separated by semicolons (;), and each token represents specific information.

- The first token is the location to insert the remaining five tokens at runtime. For example, `IFX/PaySvcRq/PmtAddRq`.
- The second token is the name of the data transformation map for mapping the external request integration object indicated by ERq to the internal request integration object indicated by IRq. For example, `PmtAddRq_ERqIRqMapIn`.
- The third token is the name of the data transformation map for mapping the internal response integration object IRs to the external response integration object ERs. For example, `PmtAddRq_IRqERqMapOut`.
- The fourth token is the external request integration object. For example, `FINS IFX Tutorial PmtAddRq IntgObj`.
- The fifth token is the internal response integration object. For example, `FINS IFX Tutorial Payment Add Internal IntgObj`.
- The sixth token is the action for the FINS IFX Transaction Manager service to perform for incoming messages. This string should match a user property defined on the FINS IFX Transaction Manager service that defines which business service and method the transaction manager should invoke. For example, `IXMLOperation_ADD`.

The data transformation map names must be used when configuring the transformation maps. For details, see "Configuring the Data Transformation Maps" on page 3-92. The map names have to be unique and you need to modify the dispatcher map entries to reflect the new name. The same principle applies to all the tokens.

To modify the dispatcher map

1. Display the user properties of the dispatcher map, as described in the previous procedure.
2. In the user property IFX/PaySvcRs/PmtAddRs, delete IXML Operation_ADD, the last token in the value.

   - Original value = IFX/PaySvcRs/PmtAddRs;PmtAddRs_ERsIRsMapIn;PmtAddRs_IRqERqMapOut;FINS IFX Tutorial PmtAddRs IntgObj;FINS IFX Tutorial Payment Add Internal IntgObj; IXMLOperation_ADD
   - Modified value = IFX/PaySvcRs/PmtAddRs;PmtAddRs_ERsIRsMapIn;PmtAddRs_IRqERqMapOut;FINS IFX Tutorial PmtAddRs IntgObj;FINS IFX Tutorial Payment Add Internal IntgObj;

   By deleting the operation portion of the dispatcher map entry, the outbound incoming transaction manager will not invoke any business service for further processing.

Compiling the Integration Objects

After you create the integration objects, you compile them into the repository file for use by the connector.

To compile the integration objects

1. Make sure the following integration objects are created in the FINS IFX Tutorial project:
   - FINS IFX Tutorial DispMap
   - FINS IFX Tutorial Envelope
   - FINS IFX Tutorial PmtAddRq IntgObj
   - FINS IFX Tutorial PmtAddRs IntgObj
   - FINS IFX Tutorial Payment Add Internal IntgObj
2. From Siebel Tools, choose Tools > Compile Projects.

   The Object Compiler dialog box appears.
3. Choose the project, browse to the repository file for the project, or create a new one, and then click Compile.

NOTE: Compile all the integration objects and the dispatcher map created by the wizard into your .srf. Make sure you migrate your new integration objects to the same database used by your client. You also need to copy your newly compiled .srf to the correct object directory of the same server used by your clients.

Configuring the Connector Components

Siebel Connector for IFX XML provides four prebuilt business services that you can configure for your specific use:

- FINS IFX XML Transaction Manager
- FINS IFX XML Data Transformation Engine
- FINS IFX XML Converter
- FINS IFX XML Dispatcher

Each business service has its own user properties. The values of these user properties are decided by configuration time. However, you can also override those values in the workflow by entering a run-time value. The meanings of the user properties are described in Chapter 2, "Siebel Connector for IFX XML." This section shows the configuration of the user properties for each of these business services.

FINS IFX XML Transaction Manager

The example uses a duplicated and modified FINS IFX XML Transaction Manager.

To create the example transaction manager

1. In the FINS IFX Tutorial project, right click FINS IFX XML Transaction Manager record, and select Copy Record from the pop-up menu.
2. Fill in the following values for the new business service record, if necessary:
   - Name = FINS IFX Tutorial Transaction Manager
   - Project = FINS IFX Tutorial
   - Display Name = FINS IFX Tutorial Transaction Manager

Several prebuilt operations have been defined in the transaction manager. These operations are sufficient to support most needs in the Siebel Connector for IFX XML. It is recommended that you not change these values unless you want to add new operations.

Table 27. User Properties for the FINS IFX XML Transaction Manager

Operation Name	Value
BlankIOI	FINS IFX XML Extension/CreateIntObjInstance/
QuickQuery	FINS IFX XML Extension/QuickQuery/#XMLHierarchy;BC Name;ReturnField;SearchSpec;
SAQuery	EAI Siebel Adapter/Query/
SARowIdQuery	EAI Siebel Adapter/Query/PrimaryRowId;!SiebelMessage;
SASynchronize	EAI Siebel Adapter/Synchronize/
SAUpsert	EAI Siebel Adapter/Upsert/

Following are examples of how the values in Table 27 are interpreted:

- BlankIOI means the operation will create an integration object instance that contains all the elements defined in the integration object, but with empty values.
- QuickQuery means the operation will retrieve the value of < ReturnField > of the first record of the search result based on < SearchSpec > against the < BC Name > business component.
- SAQuery means the operation will execute the EAI Siebel Adapter's Query method.
- SAUpsert means the operation will execute the EAI Siebel Adapter's Upsert method.

The basic format for the value entry is as follows:

- Service/Method/Argument;Argument;
- /Method/Argument;Argument;

- Service, method, and argument are separated by a slash (/).
- Each argument ends with a semicolon (;).
- The default Service name is EAI Siebel Adapter.
- The default argument name is SiebelMessage.

FINS IFX XML Data Transformation Engine

The example uses a duplicated and modified FINS IFX XML DTE.

To create the example DTE

1. In the FINS IFX project, right click FINS IFX XML DTE record, and select Copy Record from the pop-up menu.
2. Fill in the following values for the new business service record, if necessary:
   - Name = FINS IFX Tutorial DTE
   - Project = FINS IFX Tutorial
   - Display Name = FINS IFX Tutorial DTE

FINS IFX XML Converter

The example uses a duplicated and modified FINS IFX XML Converter.

To create the example converter

1. In the FINS IFX project, right click FINS IFX XML Converter record, and select Copy Record from the pop-up menu.
2. Fill in the following values for the new business service record, if necessary:
   - Name = FINS IFX Tutorial Converter
   - Project = FINS IFX Tutorial
   - Display Name = FINS IFX Tutorial Converter

3 Set the user property values according to the following table. These values will appear in the pre-header section of an IFX message.

Name	Value
XMLEnvIntObjectName	FINS IFX Tutorial Envelope
EscapeNames	TRUE
PI_Parameter:version	1.0.1
PI_Parameter:newfileuid	(default empty)
PI_Parameter:oldfileuid	(default empty)

FINS IFX XML Dispatcher

The example uses a duplicated and modified FINS IFX XML Dispatcher.

To create the example dispatcher

1 In the FINS IFX Tutorial project, right click FINS IFX XML Dispatcher record, and select Copy Record from the pop-up menu.

2 Fill in the following values for the new business service record, if necessary:

- Name = FINS IFX Tutorial Dispatcher
- Project = FINS IFX Tutorial
- Display Name = FINS IFX Tutorial Dispatcher

3 Fill in names of dispatcher map and envelope integration objects that are created by FINS IFX Wizard.

Name	Value
DispatcherMapName	FINS IFX Tutorial DispMap
XMLEnvIntObjectName	FINS IFX Tutorial Envelope

NOTE: After configuring each business service, you need to compile the new business service definition into the application repository file. The procedure is the same as compiling an integration object. For instructions, see "Compiling the Integration Objects" on page 3-86.

Configuring the Data Transformation Maps

Configuring the integration objects associates the fields in an internal integration object with the message elements in an external integration object. The result is the creation of the DTE map that will be used by the data transformation engine.

All entries created by the wizard are stored in the Integration Object User Properties of the Dispatcher Map.

In the example, there are four maps that need to be configured to have a complete outbound/inbound transaction route available. Each one can be found in the user properties entry in the IFX Tutorial DispMap dispatcher map integration object.

The integration object for the server entry is IFX/PaySvcRq/PmtAddRq, and it has two maps, as follows:

- PmtAddRq_ERqIRqMapIn (server receiving incoming request)
- PmtAddRq_IrsERsMapOut (server sending out outgoing response)

The integration object for the client is IFX/PaySvcRs/PmtAddRs, and it has two maps, as follows:

- PmtAddRs_IRqERqMapOut (client sending a outgoing request)
- PmtAddRs_ErsIRsMapIn (client receiving incoming response)

If you wish, you can change the map name in the Dispatcher Map list, then use the new name for the DTE map.

The tutorial example shows the steps for configuring two maps, used by the outbound cycle. For detailed information, see the following Siebel EAI documents: the chapter on creating and using dispatch rules *in Siebel Financial Services eBusiness Application Integration Guide, and the chapter on* data mapping and the data mapper in *Business Processes and Rules: Siebel eBusiness Application Integration Volume IV*.

Configuring an Outbound Request DTE Map

This section describes how to configure the data transformation engine map so that a payment add service request in Siebel hierarchy can be mapped to a PmtAddRq in IFX hierarchy.

- **Source Object.** For an outgoing message, the source object is the internal integration object; and for an incoming message the source object is the external integration object.
- **Target Object.** For an outgoing message, the target object is the external integration object; and for an incoming message, the target object is the internal integration object.

To configure the DTE map

1. Start Siebel Financial Services 7.0.
2. Open the Integration Administration view.

   From the application-level menu, choose View > Site Map > Integration Administration.
3. Click Data Map Editor, and in the Integrated Object Map applet, create a new map.
   - **Name.** This name must be the same as the DTE map name created by the wizard and stored in the dispatcher map list. For the example, enter PmtAddRs_IRqERqMapOut.
   - **Source Object Name.** For an outgoing message, the source object is the internal integration object. For the example, enter FINS IFX Tutorial Payment Add Internal IntgObj.
   - **Target Object Name.** For an outgoing message, the target object is the external integration object. For the example, enter FINS IFX Tutorial PmtAddRq IntgObj.

4 Create new records in the Integration Component Map applet and the Integration Field Map applet, using the values in the following tables.

The tables are shown in the next section, "Integration Component and Field Map Records" on page 3-94.

5 After creating the records:

- In the Integration Object Map applet, click the Validate button to validate the map you have just created.
- In the Integration Object Map applet, click the Save button to save the map you have just created.

Integration Component and Field Map Records

The following tables show the values for the set of records you create in the Integration Component Map applet and the Integration Field Map applet. They apply to Step 4 on page 3-94.

- Table 28 shows the values for the Container integration component map record.

Table 28. Container Integration Component Map Record

Field	Value
Name	Container
Source Component Name	Service Request
Target Component Name	SiebelFINSIXMLContainer

- Table 29 shows the values for the Payee Id integration component map record. Table 30 shows the values for the Payee Id field map record.

Table 29. Payee Id Integration Component Map Record

Field	Value
Name	Payee Id
Source Component Name	FIN Service Request Payment Request
Target Component Name	CustPayeeId
Parent Component Map Name	RemitInfo

Table 30. Payee Id Integration Field Map Record

Field	Value
Source Expression	[Payee Id]
Target Field Name	PCDATA_CustPayeeId

- Table 31 shows the values for the Amt integration component map record. Table 32 shows the values for the Amt field map record.

Table 31. Amt Integration Component Map Record

Field	Value
Name	Amt
Source Component Name	FIN Service Request Payment Request
Target Component Name	Amt
Parent Component Map Name	CurAmt

Table 32. Amt Integration Field Map Record

Field	Value
Source Expression	[Amount Transfer]
Target Field Name	PCDATA_Amt

- Table 33 shows the values for the Currency integration component map record. Table 34 shows the values for the Currency field map record.

Table 33. Currency Integration Component Map Record

Field	Value
Name	Currency
Source Component Name	FIN Service Request Payment Request
Target Component Name	CurCode
Parent Component Map Name	CurAmt

Table 34. Currency Integration Field Map Record

Field	Value
Source Expression	[Currency]
Target Field Name	PCDATA_CurCode

- Table 35 shows the values for the AcctId integration component map record. Table 36 shows the values for the AcctId field map record.

Table 35. AcctId Integration Component Map Record

Field	Value
Name	AcctId
Source Component Name	FIN Service Request Payment Request
Target Component Name	AcctId
Parent Component Map Name	CardAcctIdFrom

Table 36. AcctId Integration Field Map Record

Field	Value
Source Expression	[From Account Number]
Target Field Name	PCDATA_AcctId

- Table 37 shows the values for the AcctType integration component map record. Table 38 shows the values for the AcctType field map record.

Table 37. AcctType Integration Component Map Record

Field	Value
Name	AcctType
Source Component Name	FIN Service Request Payment Request
Target Component Name	AcctType
Parent Component Map Name	CardAcctIdFrom

Table 38. AcctType Integration Field Map Record

Field	Value
Source Expression	[Financial Account Type]
Target Field Name	PCDATA_AcctType

- Table 39 shows the values for the DueDt integration component map record. Table 40 shows the values for the DueDt field map record.

Table 39. DueDt Integration Component Map Record

Field	Value
Name	DueDt
Source Component Name	FIN Service Request Payment Request
Target Component Name	DueDt
Parent Component Map Name	PmtInfo

Table 40. DueDt Integration Field Map Record

Field	Value
Source Expression	[Date of Transfer]
Target Field Name	PCDATA_DueDt

- Table 41 shows the values for the DupChkOverride integration component map record. Table 42 shows the values for the DupChkOverride field map record.

Table 41. DuplicateCheck Integration Component Map Record

Field	Value
Name	DupChkOverride
Source Component Name	FIN Service Request Payment Request
Target Component Name	DupChkOverride
Parent Component Map Name	Container

Table 42. DuplicateCheck Integration Field Map Record

Field	Value
Source Expression	1
Target Field Name	PCDATA_DupChkOverride

- Table 43 shows the values for the PmtInfo integration component map record.

Table 43. PmtInfo Integration Component Map Record

Field	Value
Name	PmtInfo
Source Component Name	FIN Service Request Payment Request
Target Component Name	PmtInfo
Parent Component Map Name	Container

- Table 44 shows the values for the RemitInfo integration component map record.

Table 44. RemitInfo Integration Component Map Record

Field	Value
Name	RemitInfo
Source Component Name	FIN Service Request Payment Request
Target Component Name	RemitInfo
Parent Component Map Name	PmtInfo

- Table 45 shows the values for the CurAmt integration component map record.

Table 45. CurAmt Integration Component Map Record

Field	Value
Name	CurAmt
Source Component Name	FIN Service Request Payment Request
Target Component Name	CurAmt
Parent Component Map Name	RemitInfo

- Table 46 shows the values for the CardAcctIdFrom integration component map record.

Table 46. CardAcctIdFrom Integration Component Map Record

Field	Value
Name	CardAcctIdFrom
Source Component Name	FIN Service Request Payment Request
Target Component Name	CardAcctIdFrom
Parent Component Map Name	PmtInfo

Configuring an Outbound Response DTE Map

The response portion of an outbound message cycle handles the response message returned from the external application. For this example, since the example does not insert a record or update a Siebel database with the returned values, the response DTE map does not need to contain integration component maps.

For detailed information, see the following Siebel EAI documents: the chapter on creating and using dispatch rules *in Siebel Financial Services eBusiness Application Integration Guide, and the chapter on* data mapping and the data mapper in *Business Processes and Rules: Siebel eBusiness Application Integration Volume IV*.

To configure the DTE map

1. Start Siebel Financial Services 7.0.
2. Open the Integration Administration View.

   From the Application-level menu, choose View > Site Map > Integration Administration screen.
3. Click Data Map Editor, and in the Integrated Object Map Applet, create a new map.

   - **Name.** This name must be the same as the DTE map name created by the wizard and stored in the dispatcher map list. For the example, enter PmtAddRs_ERsIRsMapIn.
   - **Source Object Name.** For an incoming message the source object is the external integration object. For the example, enter FINS IFX Tutorial PmtAddRs IntgObj.
   - **Target Object Name.** For an incoming message, the target object is the internal integration object. For the example, enter FINS IFX Tutorial Payment Add Internal IntgObj.
4. After creating the record:

   - In the Integration Object Map applet, click the Validate button to validate the map you have just created.
   - In the Integration Object Map applet, click the Save button to save the map you have just created.

Configuring the Workflow Process

The example in this section shows how to create a workflow to handle an outbound IFX XML request and response message pair.

Here are the distinctions to keep in mind:

- **Outbound.** The message starts in a Siebel client. Siebel application sends a request to the external data source and receives a response from the external data source.
- **Inbound.** The message starts in an external data source. Siebel receives a request from the external data source and sends a response to the external data source.

Whether outbound or inbound, the cycle consists of a request and response message pair.

Creating a Workflow for the IFX XML Request Message

The following procedures show how to construct a converter to handle a request message.

1. Start Siebel Financial Services 7.0
2. Open the Siebel Workflow Administration.
3. Create a new record in the Workflow Process View, using the following values:
   - Name = FINS IFX Tutorial Outbound Connector
   - Business Object = Service Request

   The other values are optional.
4. Click the Process Designer tab.

   The Process Designer provides a blank working space onto which you will drag the step symbols and connectors that create the workflow.

   For complete details about using this working space, and information about workflows in general, see *Siebel Workflow Administration Guide.*

5 Drag Start, Wait, Stop, and Business Service steps onto the work space, and name them appropriately, using connector arrows to connect them.

Continue until you have created a workflow with the required components. Figure 6 shows the structure of the finished workflow.

6 Double-click the Start step to display the Start and Next Steps applet, and set the following value in the single Next Steps record:

- Type = Default

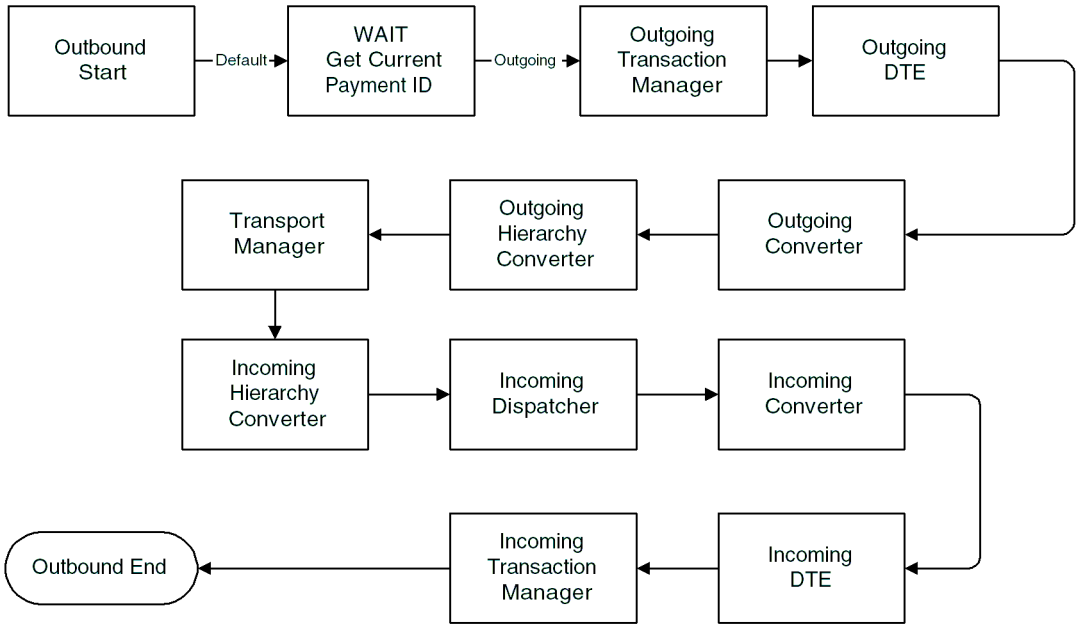


Figure 6. IFX Add Payment Outbound Workflow

Adding Process Property Values

You need to create Process Property values for the workflow, to be used in later configurations.

To create process properties

1. Click the Process Properties tab.
2. Enter the values shown in the following table.

Name	Data Type
End of Data	String
IFX Document Request	String
IFX Document Response	String
OP Outgoing TransMgr	String
PropSet Out Incoming Converter	Hierarchy
PropSet Out Incoming DTE	Hierarchy
PropSet Out Incoming Dispatcher	Hierarchy
PropSet Out Incoming Hierarchy Converter	Hierarchy
PropSet Out Incoming TransMgr	Hierarchy
PropSet Out Outgoing Converter	Hierarchy
PropSet Out Outgoing DTE	Hierarchy
PropSet Out Outgoing TransMgr	Hierarchy
SearchSpec	String
SearchSpec%1	String
Timed Out	String

Configuring the Wait Step

The Wait step can be configured to perform an action after the workflow is initiated. When the action is taken, the remaining steps will then be performed. In this example, the Wait step is configured to retrieve the Row_ID of the current record.

To configure the Wait step

1. Double click the Wait step of FINS IFX Tutorial Outbound Connector workflow.
2. In the Wait applet, enter the following value:
   - Name = Get Current Payment Id
3. Leave the Input Arguments applet empty (that is, with no records).
4. Add output arguments in the Output Arguments applet (shown in the table below), and then click the Return to Designer button.

Property Name	Type	Value	Business Component Name
SearchSpec	Literal	[FIN Service Request Payment Request.Payment Id] = %1	
Search Spec%1	Expression	[Id]	Fin Service Request Payment Request

Configuring the FINS IFX XML Transaction Manager

The first outbound component is the transaction manager.

To configure the transaction manager

1. Name the transaction manager appropriately (it must be unique within the workflow), and select the type.

   The example uses:

   - Name = Outgoing Transaction Manager
   - Type = Business Service

2. Choose the type of business service from the multi-value group (MVG) window.

   For this component, choose the FINS IFX Tutorial Transaction Manager.

3. Choose the method.

   The example uses the Execute Outbound method because the data is being sent out from the Siebel data source.

4 Establish the input and output arguments.

As for all Siebel business services, create a new record (CTRL + N) for a new argument. Select from the available input arguments, and type in any that do not appear in the list. Add all required arguments first, then go on to any optional arguments. See Chapter 2, "Siebel Connector for IFX XML," for input and output specifications.

When you have finished setting the input and output arguments, click the Return to Designer button.

Following are the input argument settings for the example transaction manager configuration.

Input Argument	Type	Value	Property Name
DispatcherMapName	Literal	FINS IFX Tutorial DispMap	
IXMLMapPath	Literal	IFX/PaySvcRs/ PmtAddRs	
PrimaryRowId	Process Property		Object Id
SiebelFINSOperationOut	Literal	SAQuery	
SiebelFINSRespIntObjName	Literal	FINS IFX Tutorial Payment Add Internal IntgObj	
SearchSpec	Process Property		SearchSpec
SearchSpec%1	Process Property		SearchSpec%1

Following are the output argument settings for the example transaction manager configuration.

Property Name	Type	Value	Output Argument
PropSet Out Outgoing Transmgr	Output Argument		XML Hierarchy

Configuring the FINS IFX XML Data Transformation Engine

The second component is the data transformation engine (DTE).

To configure the DTE

1. Name the DTE appropriately (it must be unique within the workflow).

   The example uses:

   - Name = Outgoing DTE
   - Type = Business Service

2. Choose the type of business service from the multi-value group (MVG) window.

   For this component, choose the FINS IFX Tutorial DTE.

3. Choose the method.

   The example uses the Transform to External Hierarchy method because the data is moving from a Siebel internal system to an external system.

4. Set the input and output arguments.

   These arguments include the DTE map name, created during the configuration of the internal and external integration objects.

   Following are the input argument settings for the example DTE configuration.

Input Argument	Type	Value	Property Name
XML Property Set	Process Property		PropSet Outgoing TransMgr

   Following are the input argument settings for the example DTE configuration.

Property Name	Type	Value	Output Argument
PropSet Out Outgoing DTE	Output Argument		XML Property Set

Configuring the FINS IFX XML Converter

The third component is the converter.

To configure the converter

1. Name the converter appropriately (it must be unique within the workflow).

   The example uses:

   - Name = Outgoing Converter
   - Type = Business Service

2. Choose the type of business service from the multi-value group (MVG) window.

   For this component, choose the FINS IFX Tutorial Converter.

3. Choose the method.

   The example uses the PropSetToXMLPropSet method because the converter is converting a property set from the DTE into a standard XML property set.

4 Set the input and output arguments.

Following are the input argument settings for the example converter configuration.

Input Argument	Type	Value	Property Name
Client Application Name	Literal	Siebel FINS Application	
Client Application Organization	Literal	Siebel FINS	
Client Application Version	Literal	7.0	
Date Output Format	Literal	YYYY-MM-DD	
Enable Data Formatting	Literal	true	
IFX Application Business Service Name	Literal	FINS IFX XML Extension	
Is Client	Literal	true	
Is Log Out	Literal	true	
Service Provider Name	Literal	Partner IFX Middleware	
XMLEnvIntObjectName	Literal	FINS IFX Tutorial Envelope	
XML Property Set	Process Property		PropSet Out Outgoing DTE

Following are the output argument settings for the example converter configuration.

Property Name	Type	Value	Output Argument
PropSet Converter	Output Argument		XML Property Set

Configuring an XML Hierarchy Converter

The fourth component is a hierarchy converter. For information about Siebel hierarchy converters, see *Siebel Workflow Administration Guide*.

To configure an XML hierarchy converter

1. Name the hierarchy converter appropriately (it must be unique within the workflow).

   The example uses:

   - Name = Outgoing Hierarchy Converter
   - Type = Business Service

2. Choose the type of business service from the multi-value group (MVG) window.

   For this component choose the XML Hierarchy Converter.

3. Choose the method.

   This example uses the XML Hierarchy to XML Document method because, for an outgoing message, the final conversion is from an XML hierarchy to an XML document that can be accepted by any XML-compliant converter at the external location.

4 Set the input and output arguments, XML Hierarchy name, and XML document name.

Following are the input argument settings for the example hierarchy converter configuration.

Input Argument	Type	Value	Property Name
Escape Name	Literal	false	
XML Header Text	Literal	< !DOCTYPE IFX SYSTEM "http://*IFXWebSite*/IFX101.dtd" > Replace *IFXWebSite* with a valid connection string that connects to the IFX101.dtd.	
XML Hierarchy	Process Property		PropSet Out Outgoing Converter

Following are the output argument settings for the example hierarchy converter configuration.

Property Name	Type	Value	Output Argument
IFX Document Request	Output Argument		XML Document

Configuring a Transport Mechanism

The fifth component is a transport mechanism. The example uses the HTTP transport. For information about transport mechanisms supported by Siebel Systems, see *Transports and Interfaces: Siebel eBusiness Application Integration Volume III*.

To configure an HTTP transport

1. Name the transport component appropriately (it must be unique within the workflow).

   The example uses:

   - Name = Transport Manager
   - Type = Business Service

2. Choose the type of business service from the multi-value group (MVG) window.

   For this component, choose the IBC Transport Manager.

3. Choose the method.

   This example uses the Send and Receive method, so that the adapter will wait for a specified time for the external system to process the request and send a response.

4 Set the input and output arguments, including the Physical Queue Name, the Queue Manager Name, and the Message Text.

Following are the input argument settings for the example HTTP transport configuration. Notice that the data flow within the connector changes direction at this point from outgoing to incoming.

Input Argument	Type	Value	Property Name
Message Text	Process Property		IFX Document Request
Connect Info	Literal	http://*IFXMiddleware*/ Replace *IFXMiddleware* with a valid connection string from which you can obtain IFX service.	
HTTP User Agent Header Value	Literal	ifx Use all lower-case letters.	

Following are the output argument settings for the example HTTP transport configuration.

Property Name	Type	Output Argument
End of Data	Output Argument	End of Data
IFX Document Response	Output Argument	Message Text
Timed Out	Output Argument	Timed Out

This step completes the request portion of the workflow, and it begins the response portion of the workflow. The outbound Siebel Connector for IFX XML can be used as the basis for any workflow that is used to send an outbound request message and wait for a response message.

Alternative to the Transport Mechanism Business Service Step

If you do not have an IFX Service Provider, you can replace this business service step with two business service steps:

- A business service step that writes the IFX request document to a file.
- A business service step that reads the corresponding IFX response document from a file.

You may want to use this alternative during development, so that you can work without using an IFX server or middleware. Save all request and response message pairs into files to test all possibilities locally. A single IFX document should be saved in each file. The number of files depends on how many IFX documents you want to maintain.

Configuring the Siebel Connector for IFX XML Response

The response portion of the cycle is the partner to the outgoing portion in an outbound cycle.

Configuring an XML Hierarchy Converter

The first connector component in the response sequence is a hierarchy converter. For information about Siebel hierarchy converters, see *Siebel Workflow Administration Guide*.

To configure an XML hierarchy converter

1. Name the hierarchy converter appropriately (it must be unique within the workflow).
   - Name = Incoming Hierarchy Converter
   - Type = Business Service
2. Choose the type of business service from the multi-value group (MVG) window.

   For this component, choose the XML Hierarchy Converter.

3 Choose the method.

This example uses the XML Document to XML Hierarchy method because, for an incoming message, the conversion is from an XML document to an XML hierarchy that can be used by the FINS IFX XML Dispatcher.

4 Set the input and output arguments, XML Hierarchy name, and XML document name.

Following are the input argument settings for the example hierarchy converter incoming configuration.

Input Argument	Type	Value	Property Name
XML Document	Process Property		IFX Document Response

Following are the output argument settings for the example hierarchy converter incoming configuration.

Property Name	Type	Value	Output Argument
PropSet Out Incoming Hierarchy Converter	Output Argument		XML Hierarchy

Configuring the FINS IFX XML Dispatcher

The second connector component in the response sequence is the dispatcher.

To configure the dispatcher

1 Name the dispatcher appropriately (it must be unique within the workflow).

The example uses:

- Name = Incoming Dispatcher
- Type = Business Service

2 Choose the type of business service from the multi-value group (MVG) window.

For this component, choose the FINS IFX Tutorial Dispatcher.

3 Choose the method.

The example uses the Dispatch Message method.

4 Set the input and output arguments.

Following are the input argument settings for the example dispatcher incoming configuration.

Input Argument	Type	Value	Property Name
DispatcherMapName	Literal	FINS IFX Tutorial DispMap	
XMLEnvIntObjectName	Literal	FINS IFX Tutorial Envelope	
XML Property Set	Process Property		PropSet Out Incoming Hierarchy Converter

Following are the output argument settings for the example dispatcher incoming configuration.

Property Name	Type	Value	Output Argument
PropSet Out Incoming Dispatcher	Output Argument		XML Property Set

Configuring the FINS IFX XML Converter

The third response component is the converter.

To configure the converter

1. Name the converter appropriately (it must be unique within the workflow).

   The example uses:

   - Name = Incoming Converter
   - Type = Business Service

2. Choose the type of business service from the multi-value group (MVG) window.

   For this component, choose the FINS IFX Tutorial Converter.

3. Choose the method.

   The example uses the XMLPropSetToPropSet method, because the converter is converting an XML property set from the dispatcher into a Siebel-hierarchy-based property set.

4 Set the input and output arguments.

Following are the input argument settings for the example converter incoming configuration.

Input Argument	Type	Value	Property Name
Enable Data Formatting	Literal	True	
IFX Application Business Service Name	Literal	FINS IFX XML Extension	
XMLEnvIntObjectName	Literal	FINS IFX Tutorial Envelope	
XML Property Set	Process Property		PropSet Out Incoming Dispatcher

Following are the output argument settings for the example converter incoming configuration.

Property Name	Type	Value	Output Argument
PropSet Out Incoming Converter	Output Argument		XML Property Set

Configuring the FINS IFX XML Data Transformation Engine

The fourth response component is the data transformation engine (DTE).

To configure the DTE

1 Name the DTE appropriately (it must be unique within the workflow).

The example uses:

- Name = Incoming DTE
- Type = Business Service

2 Choose the type of business service from the multi-value group (MVG) window.

For this component, choose the FINS IFX Tutorial DTE.

3 Choose the method.

The example uses the Transform to Siebel Hierarchy method because the data is moving from an external system to a Siebel internal system.

4 Set the input and output arguments.

Following are the input argument settings for the example DTE response configuration.

Input Argument	Type	Value	Property Name
XML Property Set	Process Property		PropSet Out Incoming Converter

Following are the input argument settings for the example DTE response configuration.

Property Name	Type	Value	Output Argument
PropSet Out Incoming DTE	Output Argument		XML Property Set

Configuring the FINS IFX Transaction Manager

The fifth response component is the transaction manager.

To configure the transaction manager

1 Name the transaction manager appropriately (it must be unique within the workflow).

The example uses:

- Name = Incoming Transaction Manager
- Type = Business Service

2 Choose the type of business service from the multi-value group (MVG) window.

For this component, choose the FINS IFX Tutorial Transaction Manager.

3 Choose the method.

The example uses the Execute Transaction method because the data is being received from the external data source and must be delivered to the Siebel application.

4 Establish the input and output arguments.

Following are the input argument settings for the example transaction manager configuration.

Input Argument	Type	Value	Property Name
XML Property Set	Process Property		PropSet Out Incoming DTE

Following are the input argument settings for the example transaction manager configuration.

Property Name	Type	Value	Output Argument
PropSet Out Incoming TransMgr	Output Argument		XML Property Set

This step completes the outbound connector sequence.

Activating the Workflow

Now that the workflow is finished, activate the workflow.

To activate the workflow

1 Click the All Process tab.

2 Click the Activate button in the Workflow Process applet.

3 Restart the Siebel Financial Services application.

Configuring Runtime Events

Siebel supports triggering workflows based on runtime events. Using a runtime event allows you to incorporate configured workflow functions into actual applications. For complete information about runtime events, see *Siebel eEvents Management Guide*.

The example in this section describes a runtime event that sends data from the current record when it is created or modified. In the following procedure, a runtime event is defined and linked to an action that starts the example outbound workflow. It sends the active record as an IFX XML message in the example connector.

To create a runtime event

1. Start Siebel Financial Services.
2. From the Application-level menu, choose View > Site Map > Runtime Event Administration > Action Sets.
3. Create a new Action Sets record, and enter the following value:
   - Name = FINS IFX Tutorial Payment Request Add Action
4. In the More Info applet, enter the following values:
   - Action Type = BusService
   - Business Service Name = Workflow Process Manager
   - Business Service Method = RunProcess
   - Business Service Context = “ProcessName”, “FINS IFX Tutorial Outbound Connector”

     This is the name of the configured workflow that handles outbound messages.

5 Add an Events record, and enter the following values:

- Sequence = 1
- Object Type = BusComp
- Object Name = FIN Service Request Payment Request
- Event = WriteRecordNew
- Action Set Name = FINS IFX Tutorial Payment Request Add Action

6 Choose Reload Personalization from the drop-down list so the changes will take effect.

Triggering the Request Message

This procedure creates an event that will trigger the message that has been configured in this chapter.

To trigger the request message

1 After all the previous tutorial procedures have been completed, start Siebel Financial Services.

2 Display the Service Requests screen.

View > Site Map > Service Requests

3 In the Account Services applet, use the Show drop-down list to select Add Payment Schedule.

4 Add a record in the Account Services applet.

5 Click the Save button.

The new record is committed to the database.

If the transaction fails, indicated by the < StatusCode > element containing a non-zero value, FINS IFX Tutorial Converter sets the error flag. This flag causes the workflow to trigger an exception, which pops up an error message.

Additional Information A

Siebel data formats differ from the IFX data formats.

The Enable Data Formatting argument for the FINS IFX XML Converter is normally set to false, which instructs the converter to retain Siebel data formatting. The converter supports only date, time, and date/time conversions.

The default setting is preferred in two cases:

- You provide an outcall function to perform the formatting.
- Your middleware can handle the differences between Siebel and IFX formats.

If you want the Siebel Connector for IFX XML to do the formatting, you should set the argument to true, and make the following changes for the external integration component:

1. In the Integration Component field, remove the quotes in the XML Literal Value of the < Element Name > _dt:type.
2. Activate the < Element Name > _dt:type integration component field.

Figure 7 and Figure 8 illustrate the record before and after the modifications presented in these steps.

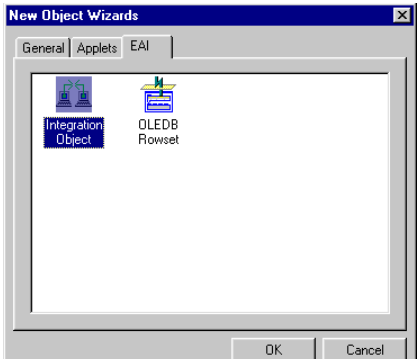


Figure 7. Before Modifications

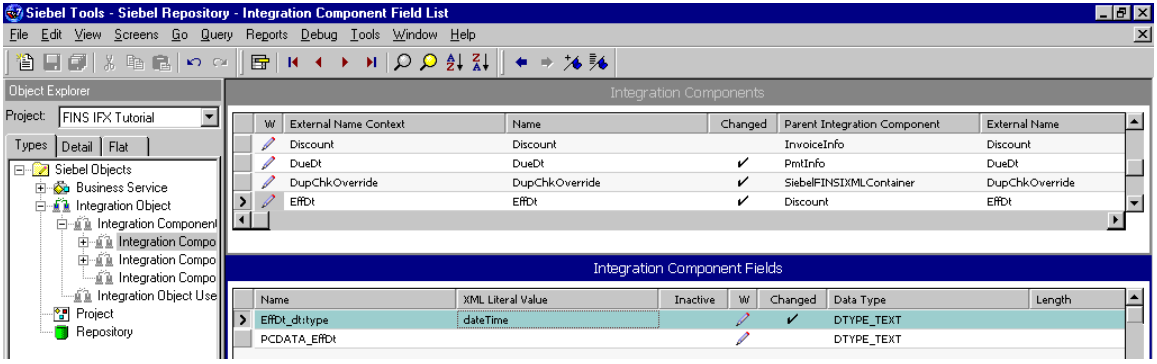


Figure 8. After Modifications

Technical Notes on Integration Components

Notice the following features of the integration components:

- Aggregate and elements are treated the same as map to integration components.
- SiebelFINSIXMLContainer is always the root integration component.
- The property Parent Integration Component and the property Cardinality are used to preserve the hierarchy and cardinality information.
- Integration component name mimics the xml tag except:
  - The dot in the xml tag is replaced with an underscore.
  - Numbers are attached to resolve naming conflicts.
- If the integration component represents an element that can have a value, it will have at least one Integration Component Field child object, which is the PCDATA.
- If the integration component represents an aggregate that cannot have a value, it will have no Integration Component Field child object.

Notice the following features of the integration objects user properties:

- Property IFX Service Name is used to provide information about what IFX Service this message belongs to. It is a required user property.
- Property Ignore Undefined XML Tags is for internal use only and is a required user property.

DTE Changes. Please note that if the underlying DTD changes, it is necessary to revise your integration object definition accordingly. Be careful of changes to commonly used aggregates, as any change will force you to regenerate external integration objects for many messages.

If you change the DTD and update your external integration objects, it is important to review your DTE maps to make sure that they are still valid and mapping the correct fields.