Siebel

# Financial Services Enterprise Application Integration Guide

January 2020



Siebel
Financial Services Enterprise Application Integration Guide

January 2020

Part Number: F12736-02

Copyright © 2020, Oracle and/or its affiliates. All rights reserved

Authors: Siebel Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display in any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

The business names used in this documentation are fictitious, and are not intended to identify any real companies currently or previously in existence.

# Contents

**Preface** i

**1 What's New in This Release** 1

What's New in Siebel Financial Services Enterprise Application Integration Guide, Siebel CRM 20.1 Update 1
What's New in Siebel Financial Services Enterprise Application Integration Guide, Siebel CRM 19.1 Update 1

**2 Overview of Siebel Financial Services Enterprise Application Integration** 3

Overview of Siebel Financial Services Enterprise Application Integration 3
About Siebel Financial Services Enterprise Application Integration 3
Components of Siebel Financial Services EAI 6
Details of Siebel Financial Services EAI 8
Siebel Financial Services EAI Connectors 11

**3 Siebel Financial Services EAI Architecture** 13

Siebel Financial Services EAI Architecture 13
Layered Architecture 13
Integration Requirements and Siebel Financial Services EAI 13
Business Process Coordination Using Workflows 14
Transport Mechanisms 15
Industry XML Connector 15

**4 Understanding Financial Services Integration Paradigms** 25

Understanding Financial Services Integration Paradigms 25
Overview 25
Design Decisions 26
Usage Models 29
Access Mechanisms 31
Application Connectors 32

**5 Financial Services EAI Business Scenarios** 33

Financial Services EAI Business Scenarios 33

Overview 33
Common Integration Scenarios 34
Data Access and Replication 35
Data Sharing Across the Enterprise 37
Data Transformation 39

6 Predefined Financial Services EAI Business Services 41
Predefined Financial Services EAI Business Services 41
Predefined Financial Services EAI Business Services 41

7 Industry XML Connector Factory APIs 43
Industry XML Connector Factory APIs 43
Industry XML Connector Factory Classes 43
Industry XML Connector Factory Classes Methods 44
Industry XML Connector Factory Method Arguments 46
Siebel Connector for Credit Card Application XML 46

# Preface

This preface introduces information sources that can help you use the application and this guide.

## Using Oracle Applications

To find guides for Oracle Applications, go to the Oracle Help Center at *http://docs.oracle.com/*.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program website*.

## Contacting Oracle

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit *My Oracle Support* or visit *Accessible Oracle Support* if you are hearing impaired.

### Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an e-mail to: *oracle_fusion_applications_help_ww_grp@oracle.com*.

# 1 What's New in This Release

## What's New in Siebel Financial Services Enterprise Application Integration Guide, Siebel CRM 20.1 Update

No new features have been added to this guide for this release. This guide has been updated to reflect only product name changes.

## What's New in Siebel Financial Services Enterprise Application Integration Guide, Siebel CRM 19.1 Update

No new features have been added to this guide for this release. This guide has been updated to reflect only product name changes.

# 2 Overview of Siebel Financial Services Enterprise Application Integration

## Overview of Siebel Financial Services Enterprise Application Integration

This chapter provides an overview of Siebel Financial Services Enterprise Application Integration.

## About Siebel Financial Services Enterprise Application Integration

Oracle's Siebel Financial Services Enterprise Application Integration platform combines a set of tools, technologies, and prebuilt functional integration components to facilitate application integration. This set of products is referred to as Siebel Financial Services Enterprise Application Integration (EAI). Siebel Financial Services EAI is an EAI solution built on Siebel EAI offered by Oracle. Siebel Financial Services EAI provides an integration solution using industry XML standards. These standards have been adopted by the industry and extended by middleware companies, as in IBM's MQSeries Financial Services Edition (MQSFSE).

Siebel Financial Services EAI is designed as a set of interfaces which interact with each other and with other components within the Siebel application. These interfaces:

- Allow configurable messages within Oracle's Siebel Tools for exchanging information using the various industry XML standards.
- Expose internal Siebel Objects to external applications.
- Take advantage of prebuilt adapters and enterprise connectors, and are compatible with third-party adapters and connectors.
- Allow prebuilt XML connectors for Siebel applications.
- Allow for comprehensive data transformation.
- Allow extension for customized XML connectors by providing the Siebel Industry XML Connector Factory API.



**Note:** For details on Siebel EAI, see *Overview: Siebel Enterprise Application Integration* .

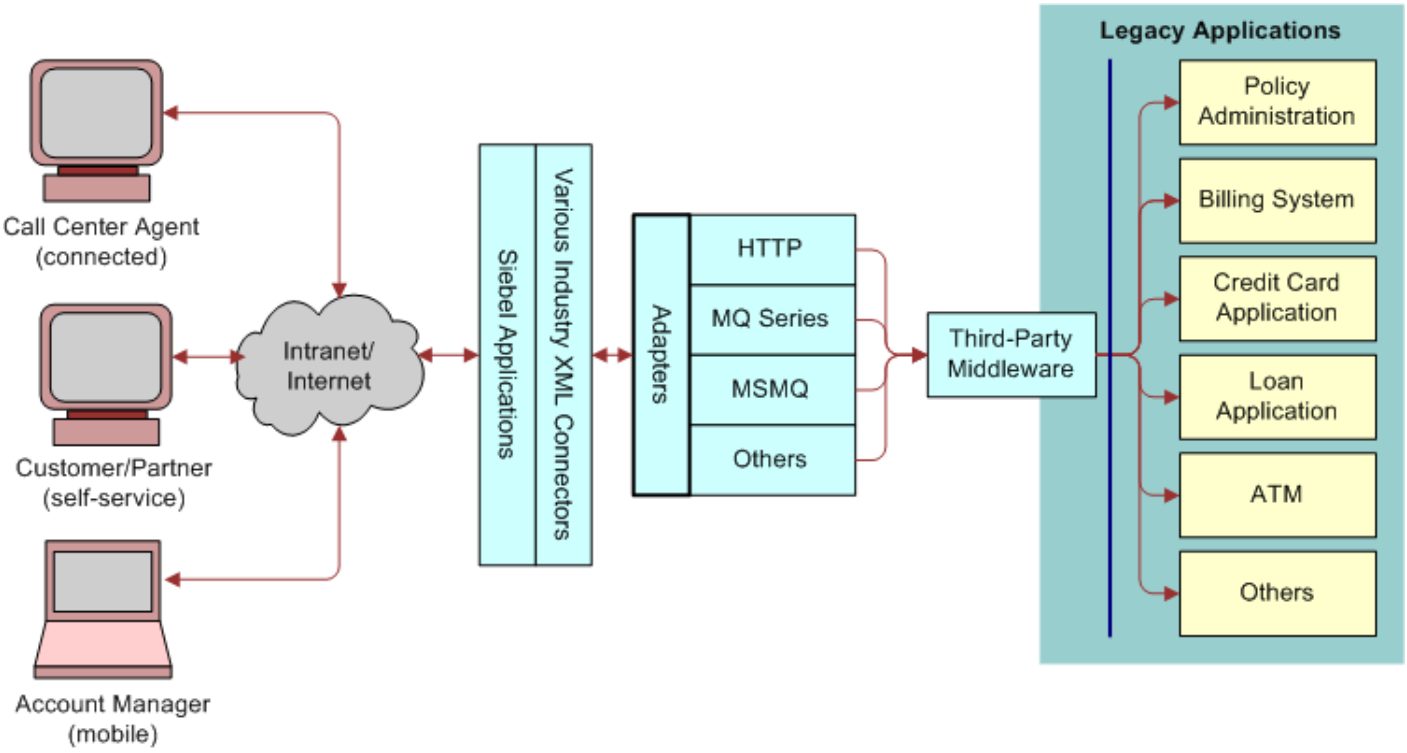
## Legacy Encapsulation

The Siebel Financial Services application can serve as either a front-end administration system or as a back-end CRM information storing system. In either case, it is necessary for the Siebel Financial Services application to integrate with one or more multi-channel back-end systems such as an Insurance Policy Administration System. To achieve this, the Siebel Financial Services application must be configured to subscribe, receive, and then filter out the necessary

information to store in the database. The goal of integration is to make the communication possible between systems that have not been designed to communicate together.

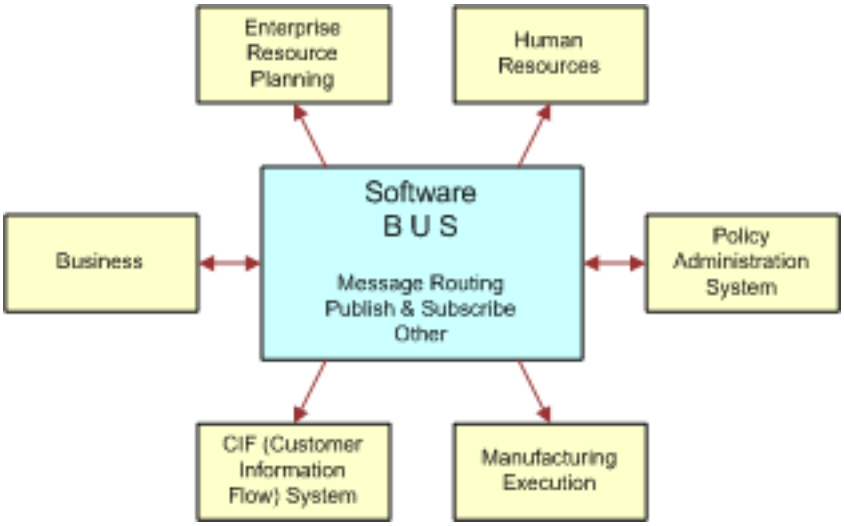
In allowing communications between systems, many different approaches have been developed. Among these, messaging architecture is widely used and supported by the industry. For example, message oriented middleware, such as IBM's MQSFSE, provides cross-referencing functionality for all the applications. MQSFSE provides the connecting avenue between source or front-end systems with target or legacy systems. Many customers in the financial services business face the same business problem of how to connect new front-end applications for CRM or Enterprise applications with existing legacy applications that were designed for a different business model. Many of these problems are addressed by a relatively small number of key business processes that are implemented in most organizations. Companies such as IBM have encapsulated their experience and expertise in the Insurance Application Architecture (IAA) organization. In a similar effort, large insurance and banking organizations introduced ACORD (Association for Operations Research Development) and IFX (Interactive Financial Exchange) standards. These standards provide the definition of key data elements of insurance and financial services, business components, and the interfaces to these components using XML messages. With the Siebel Financial Services connectors provided, the Siebel Business Applications platform generates and processes the necessary request and response expected in an encapsulated environment for legacy integration to support both inbound and outbound real-time integration between Siebel applications and other applications.

This encapsulation architecture provided in Siebel Financial Services encompasses the Siebel Call Center application as shown in the following image.




# Peer-to-Peer Integration

In peer-to-peer integration shown in the following image, multiple applications interact through a central software bus. Usually, applications use the messaging architecture and choose a specific standard. For example, the insurance industry may use the ACORD standard for the syntax and rules of the messages flowing in this information bus.




# Other Integration Strategies

Other integration strategies are applicable as well. These include master data partitioning, data sharing, presentation layer integration (screen scrapping), and others as follows:

- Data Sharing. Data Sharing is another integration strategy practiced by the financial services industry. Using this strategy you can access external data in real-time without duplicating the data. This is a very common practice due to the nature of large amounts of transaction data for a business data flow. For example, you may need to review more than one year's bank transaction records for an account or more than 10 years of a policy subscriber's credit reports that are stored in a back-end applications database.
- Presentation Layer Integration. Siebel Financial Services EAI supports client-side integration in a Web client environment. Client-side integration consists of integrating applications at the user interface level, either by just posing the screens within a portal and coordinating the content, or by screen scrapping the content of one into the other.
- Data Mapping and Transformation.Data shared between applications might not be structured or encoded in the same way. To facilitate integration of nonidentical data structures, Siebel Financial Services EAI specifies mapping and transformation rules, including filtering and structural changes. Siebel Financial Services EAI provides a framework that is ready for transforming heterogeneous data structures.
- Heterogeneous XML Message Integration. Siebel Financial Services EAI supports heterogeneous object operations for the industry XML integration solution. Heterogeneous XML Messages enhance the turnaround time between messages and decrease the overhead between messages. Instead of using one message for each request, you can use a single heterogeneous message to handle requests.
- Dispatching various XML standards. Using Siebel Financial Services EAI you can dispatch different XML standards to the target business process through the dispatching rule sets configuration and extension. Furthermore, it may also supply the necessary processing components to the target business process.
- Error Recovery. Messages sent to external applications may fail due to a variety of reasons. For example, the data sent may not be valid or sufficient. To achieve this requirement, Siebel Financial Services EAI specifies fault section elements and rollback rules to recover from the transactions executed.

# Components of Siebel Financial Services EAI

Components of Siebel Financial Services EAI allows Siebel Financial Services EAI to support information exchange and communication between Siebel applications and other applications using XML technology while obeying industry XML requirements. Siebel Financial Services EAI provides an open toolkit and framework for customers to implement integration solutions with the availability of various components, such as Siebel Industry XML Connector Factory API, Siebel Connector for ACORD XML, and Siebel Connector for IFX XML that can be used individually or together with an EAI vendor's toolkit. The following section describes the components of Siebel Financial Services EAI.

## XML and Siebel Financial Services EAI

Commercial transactions have always consisted of an exchange of goods and services, and an exchange of documents. EXtensible Markup Language (XML) allows us to implement the exchange of documents electronically. XML is essentially a platform-neutral data format that makes it possible to send messages across heterogeneous systems. Business messaging based on XML is one of the most effective ways to adapt existing application software to support external internet applications. Models are needed for integration because you must specify the structure and content of the message. Without a business content, XML is purely an empty shell.

XML provides data interchange that includes industry-specific formatting, vendor-neutral data exchange, global data exchange, language-neutral, and others. Siebel Financial Services EAI developed an infrastructure that supports the development of industry XML connectors to send, parse, and receive XML that supports different industry standards. XML assists trade to happen between different people—especially international trade because it provides common vocabularies. However, these standardized XML tags across industries are very different. Some industry groups are defining interfaces and services that could be used by industries. For example, insurance has ACORD and banking has IFX.

Siebel Financial Services EAI aimed to provide support for generic requirements across industry XML standards in order to support various Financial Services industry XML standard extensions. Siebel Financial Service EAI goal is to pinpoint the similarities among industry XML standards and thus develop and implement certain infrastructure that supports the development of industry XML connectors implement certain XML standards and messages. In the effort of providing this open infrastructure, several sets of connectors are built based on the standards emerging key roles in the financial services industry. In order to support various financial services industry standards, Siebel Financial Services EAI provides a generic connector called Industry XML Connector, which is the base connector for Siebel prebuilt connectors and Siebel Industry XML Connector Factory classes.



**Note:** Other industry XML standards can be supported by using various prebuilt Siebel connectors or implementing Siebel Industry XML Connector Factory APIs. For details, see *Industry XML Connector Factory APIs* and individual prebuilt Siebel Connector guides.

Generally the focus of companies integrating disparate applications are in two areas:

- Middleware. Some companies use middleware. You pass an XML document to the middleware and the middleware maps the data to its XML standard document structure. As a result, within an organization you can define the tags and the middleware maps between different organizations.
- Industry Standard. In the second area the tags and vocabularies are defined by organizations and companies according to industry standards. This method does not require mapping. Organizations that follow the industry standard framework can talk to each other. Also different applications within the enterprise adapting to these standards can integrate with each other.

Siebel Financial Services provides support for this direction of creating tags and vocabularies defined by industry standard organizations. Industry groups provide document type definitions (DTDs) along with tags and vocabularies to standardize data exchange within specific industries. A DTD defines the legal building blocks of an XML document by defining the document structure with a list of legal elements. It provides ways to extend the message sets to cover different business scenarios. DTD describes the document structure and provides a list of elements for a document. Groups can agree to use a common DTD for interchanging data. An application can use a standard DTD to verify that the data it receives from the outside world is valid. DTD defines messages that encapsulate an industry specific data model in depth. It contains rich sets of data exchange syntax and rules.



**Note:** None of the industry group standards have been adopted by a standards organization, such as ISO (International Standards Organization), or by W3C (World Wide Web Consortium). They are all in various stages of development.

# ACORD XML and Siebel Financial Services EAI

The ACORD XML for Property and Casualty (P&C) and Surety is an insurance industry definition of XML that contains appropriate messages for the property, casualty, and surety business sector in the insurance industry. The ACORD standard defines the required structure and format of an XML message to describe data elements and contents needed for the policy application, approval, and so on. The definition exists in the ACORD document type definition (DTD), which is incorporated by the Siebel connector to construct ACORD XML messages.

# IFX XML and Siebel Financial Services EAI

The IFX XML is the standard developed by the Interactive Financial eXchange (IFX) Forum. This organization establishes and maintains a set of XML messages for the financial industry. The IFX XML Forum publishes a DTD that allows the Siebel Connector for IFX XML to create mappings between its data and data in external databases.

# Siebel Financial Services EAI Adapters and Connectors

Siebel Financial Services EAI provides connectors to help create integrations between Siebel Financial Services applications and external applications. There are also additional connectors developed by Siebel Business Applications partners.

## Siebel Connectors

Siebel Financial Services EAI connectors provide low-level connectivity to other back-office applications such as Customer Information Integration Solutions (CIIS), but also includes the business processes used for connecting to the external applications. Used in tandem, they allow for the exchange of policy, bank account, and similar information between Siebel applications and external applications.

## Third-Party Connectors

Your organization may have a need to integrate multiple applications—such as Siebel Financial Services applications, SAP, Oracle, and so on—with one another. If so, you might need an EAI solution from one of the EAI vendors. These

vendors work closely with Oracle to develop adapters to Siebel Financial Services applications using Siebel EAI. Oracle has a validation program to make sure that these adapters work according to Oracle standards.

## Siebel Virtual Business Components

Virtual Business Components (VBC) are used to view data from an external system in Siebel applications. Using this mechanism, no replication within the Siebel database is required.

Virtual Business Components are configured in Siebel Tools and use business services to access data from an external system. VBCs use standard transports like MQSeries, HTTP, and MSMQ, along with the XML Gateway Server, to query, insert, and update data.



**Note:** For details, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

# Details of Siebel Financial Services EAI

The following section describes details of Siebel Financial Service EAI regarding upgrade, configuration, and extensibility.

## Configurable and Upgradeable Integration

Siebel Financial Services EAI can be configured both as a server and as a client. As a client, use Siebel Financial Services EAI to send out requests. You initiate requests and ask an external application to provide service. Siebel Financial Services application used as a server publishes services available within Siebel applications to receive requests from external applications. When a request is received by an Application Interface, or through a Siebel Object, it is processed and services are rendered. Siebel applications provide a comprehensive set of CRM services that can be configured as follows:

- Integration objects. Use the wizards in Siebel Tools to define integration objects. The integration object can be external, such as industry XML standard data hierarchy, or internal, such as Siebel business object hierarchy.
- Connector user properties. Connector user properties can modify default behavior, such as altering the envelope, header, or fault section of the message generated.
- Connector input arguments. Connector input arguments can take on heterogeneous messages that combine multiple integration objects that might be built from different business objects.

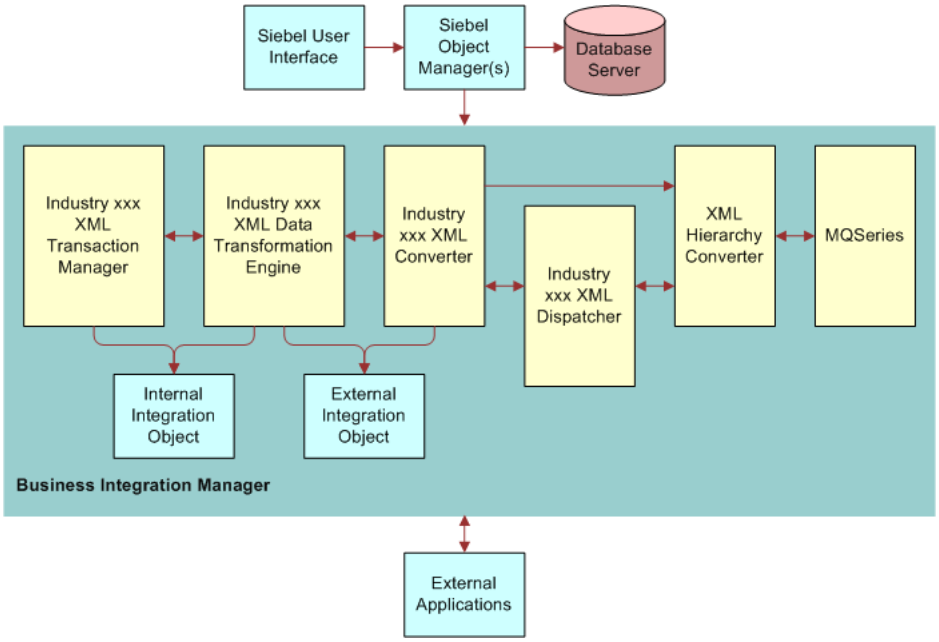
## Extensible

Siebel Financial Services EAI can be extended using standard class derivation or API implementation. You can extend the generic industry XML connector by modifying its user properties and implementing the Siebel Industry XML Connector Factory APIs to support other industry XML standards. The Industry XML Connector provides the following services that are required in most industry XML standards.

- Envelope and Header section.

- Fault or Error section.
- Body with heterogeneous command or messages.
- Data Type Formatting - Support for multiple formats. For example, Interactive Financial eXchange (IFX) has Data type in DTD.
- Data model mapping and transformation.
- Overwriting the methods available in the APIs to customize industry XML connector on the Siebel Business Application platform. The following illustrates the architecture of a customized industry XML connector.

**Note:** For details on customization solution, see *Industry XML Connector Factory APIs* .




# Declarative Integration

Siebel integration objects allow you to represent integration metadata for Siebel business objects, XML, SAP IDOCs, and SAP BAPIs as common structures 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 Oracle's Siebel eScript objects, methods, and functions, or transformed declaratively using Siebel Data Mapper. There are three types of integration object used within Siebel Financial Services EAI, Envelope Integration Object, Internal Integration Object, and External Integration Object.

Envelope Integration Object. This object presents the hierarchy of the envelope and header section of any industry XML standards.

Internal Integration Object. This object defines the Siebel data hierarchy for an external application. The internal integration object has the hierarchy of Siebel business object and business components as well as the fields. It maps to the objects in the Siebel Financial Services application and is created using Integration Object Builder in Siebel Tools. The internal integration object is required to facilitate the Industry XML Transaction Manager to package the gathered data. This service gathers the data needed for a particular message and packages it into the hierarchy defined in the internal integration object.

External Integration Object. This object defines the XML hierarchy. The external integration object maps the internal integration object components and fields to external application aggregations and fields. The Data Transformation Engine and Converter use this object to construct the external format structures.

## Data Transformation

Siebel Financial Services EAI provides a declarative data mapper (Siebel Data Mapper) to transform the data expected by Siebel applications (internal integration object) to the data received from the external system (external integration object) and the other way around. A run-time engine, the Industry XML Data Transformation Engine, implements complex domain and structure mapping. XML documents can be mapped so that one document can be transformed into another XML document with programming. For example, data in one system's LastName field can be used in another system's field named Surname. Siebel Financial Services EAI also supports transformation of heterogeneous messages.



**Note:** This type of exchange can take place within middleware, or between internal and external applications.

## Cross-Application Process Integration

The Siebel Financial Service platform supports workflow control in order to achieve process level integration between Siebel applications and external applications. You use the Siebel Workflow Designer to define the integration processes and Siebel Workflow Engine to execute and manage integration processes.

The business services you use to build the workflow process can be a prebuilt component, such as a transport adapter and the Industry XML Dispatcher, or can be custom developed at design time or run time using one of the Siebel Scripting languages.

## Siebel Database as Master for CRM

The Siebel Database is supported on multiple platforms, which allows organizations to adopt the Siebel schema as the master data schema for customers and customer relationship data. Alternatively, many financial services organizations have deployed a mainframe-based customer information file (CIF), which is used by many back-end systems. Siebel Financial Services EAI includes a number of technologies to allow using the CIF in place of, or in combination with, the equivalent structure in the Siebel system.

## Siebel Database as Integrator for Other Systems

Siebel Financial services EAI can play a subscriber role in terms of other industry data not in the CRM model that is specific to a particular industry. For example, insurance industry customers may want to store very high-level policy information within Siebel Financial Services EAI. This allows them to associate customers with certain policies and to display data from a back-end system when needed in virtual business components.

## Siebel Application as Front-End Integrator for Other Systems

You may want to query data from external systems. In this case, the search specification is captured by the Siebel Financial Services EAI infrastructure. It then sends the specification to the external system in real-time and displays the results received using Siebel connectors and virtual business components.

# Siebel Financial Services EAI Connectors

Siebel Financial Services EAI provides connectors to help create integrations between Siebel Financial Services applications and external applications.

## Siebel Connector for IFX XML

The Siebel Connector for IFX XML module extends from the Siebel Financial Services Industry XML Connector module. It is based on the IFX Forum standard for financial data exchange using XML. The Siebel Connector for IFX XML uses an IFX-compliant server, such as IBM WSBCC or another middleware application.

Siebel Connector for IFX XML provides integration between the Siebel Financial Services application and external IFX-based applications. The IFX specification supports financial communications between banks, brokerage houses (future), service providers, financial advisors (future), small businesses, and consumers. The IFX specification allows financial institutions to support customers using a broad range of client devices, 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 mobile devices such as tablets and smartphones
- Mobile telephones with data capabilities

**Note:** For details, see *Siebel Financial Services Connector for IFX XML Guide* .

## Siebel Connector for ACORD XML

The Siebel Connector for ACORD XML is an enhancement of the generic Siebel Financial Services Industry XML Connector. This connector allows Siebel applications to transfer data between Siebel applications and external ACORD-based applications. It uses XML to handle data formatting and conversion according to the ACORD forum standard.

The Siebel Connector for ACORD XML provides integration between Siebel Business Applications and other insurance application systems, such as a policy administration system that implements the ACORD XML standard. The connector supports the ACORD XML Business Message Specification for P&C Insurance and Surety, an insurance industry standard XML specification. The Siebel Connector for ACORD XML receives, parses, and processes the business operations specified in the XML message for both outbound and inbound integrations. This integration solution

offers powerful capabilities designed to meet all Property and Casualty message specification requirements. This solution also allows you to effectively harness the synergies between Siebel front office applications and ACORD-based applications. The Siebel Connector for ACORD XML allows Siebel applications to integrate with back office data and business processes for both synchronous and asynchronous transactions across applications.



**Note:** For details, see Siebel Financial Services Connector for ACORD P&C and Surety Guide.

# 3 Siebel Financial Services EAI Architecture

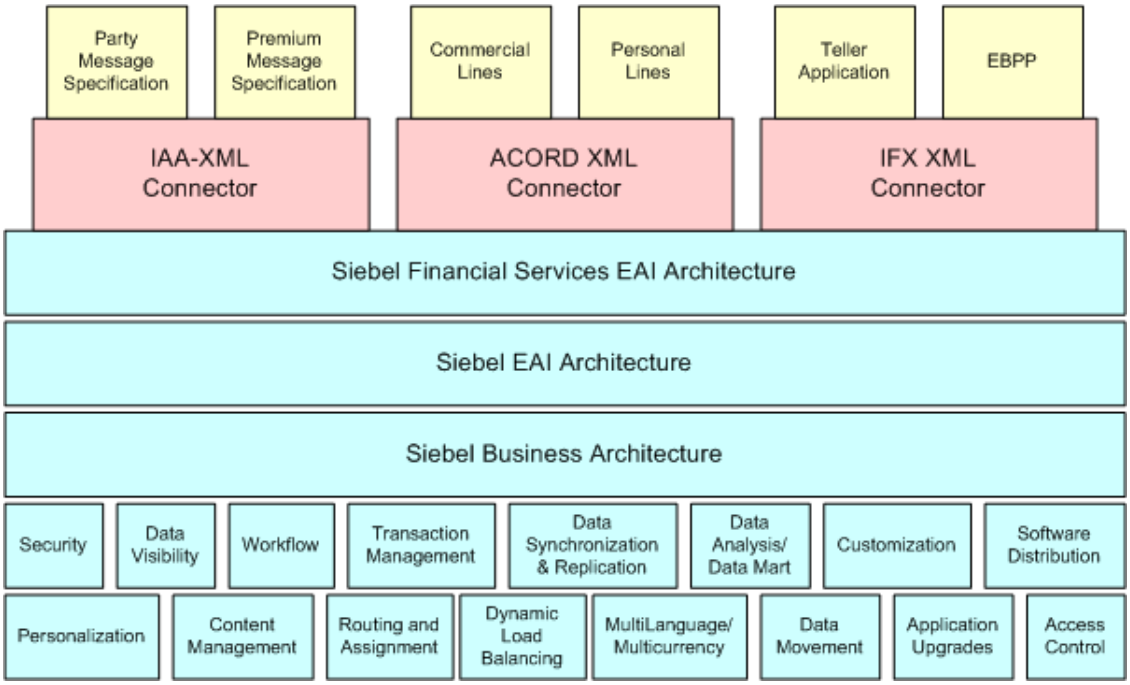
## Siebel Financial Services EAI Architecture

This chapter describes Siebel Financial Services Enterprise Application Integration architecture.

## Layered Architecture

The Siebel Financial Services Enterprise Application Integration (EAI) is an extension of the standard Siebel EAI designed to address integration approaches adopted by the vertical industry. It provides a platform for the development of specific connectors for different industry XML standards.

The following image illustrates the high-level architecture of Siebel Financial Services EAI and various industry XML connectors.



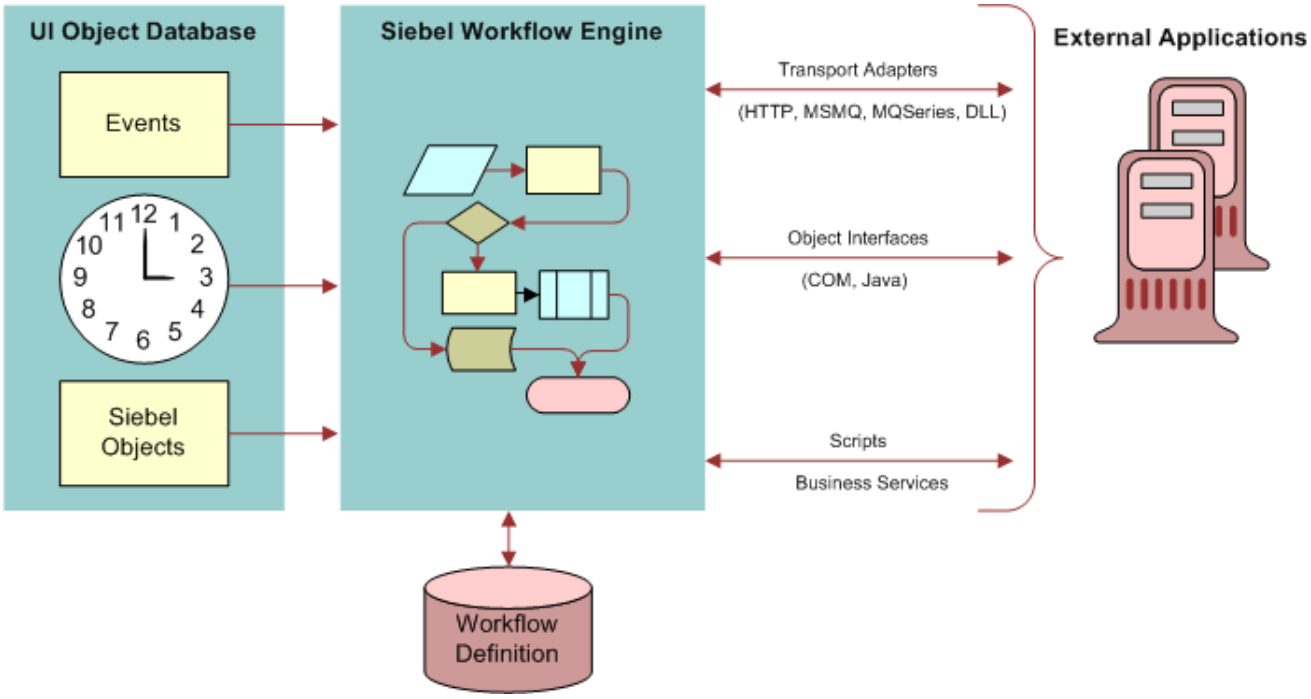

## Integration Requirements and Siebel Financial Services EAI

The following table maps real-world data and application integration requirements with their corresponding Siebel Financial Services EAI features.

| Integration Requirements | Siebel Financial Services EAI Features and Components | Notes |
|---|---|---|
| Metadata Definition<br><br>XML Metadata<br><br>External application metadata | • Integration objects based on Siebel business objects<br>• Integration objects based on external metadata<br>• FINS ACORD Wizard<br>• FINS IFX Wizard<br>• FINS OFX Wizard | In any integration project, you need a way to supply data about data, called metadata. Siebel Financial Services applications provides XML as the common format for representing external application data. In addition, Siebel Financial Services applications provides an object type in Siebel Tools called the Integration Object, as well as an Integration Object Wizard, which automates the task of creating integration objects. |
| Data Transformation<br><br>Declarative data transformation mapping<br><br>High-performance transformation engine | • Siebel Data Mapper | Siebel Financial Services EAI allows you define what type of data is to be exchanged, and allows you to export a DTD for a Siebel integration object. |
| Industry standards-based Integration Mechanism | • Interactive Financial Exchange XML (IFX XML) as a format for data exchange<br>• ACORD XML as a format for data exchange<br>• Any other industry XML standards or industry XML alike standards | See *Siebel Financial Services Connector for IFX XML Guide* , and Siebel Financial Services Connector for ACORD P&C and Surety Guide.<br><br>Also see *Industry XML Connector Factory APIs* and *Industry XML Connector*. |
| Business Process Coordination | • Workflow Process Designer<br>• Workflow Process Manager | See *Business Process Coordination Using Workflows*. |
| Transport Mechanisms (for interfacing with other technologies) | Transport Adapters:<br><br>• IBM MQ Series<br>• Microsoft MSMQ<br>• HTTP | See *Transport Mechanisms*. |
| High Volume Data Exchange | • EIM (Enterprise Integration Manager) | See *Overview: Siebel Enterprise Application Integration* . |

# Business Process Coordination Using Workflows

Connectors are composed of multiple components. These components are tied together using workflows. With different sets of configuration, you can alter generated messages and behavior. You can use the Workflow Process Designer and the Workflow Process Manager to configure your data integration business process flow. The Siebel Financial Services application provides sample workflows which you can use as a reference implementation for your data integration business processes. The following image shows one of these sample workflows from the sample database.




# Transport Mechanisms

Many companies have standardized transport solutions such as MQSeries from IBM or MSMQ from Microsoft. These transports allow the transportation of messages between one system and another. Oracle conforms to this standard by allowing you to “plug" Siebel Financial Services applications into these environments through Siebel Financial Services EAI’s comprehensive set of adapters, that include the MQSeries adapter, the MSMQ adapter, and the HTTP adapter.

Transport adapters developed by Oracle and Oracle-certified partners allow diverse applications to communicate with Siebel Financial Services applications across virtually any operating system, network, and database. The Architecture for Siebel Financial Services EAI image in *Business Process Coordination Using Workflows* shows the various available transports to and from Siebel Financial Services applications as a new customer’s information is processed. Any one or any combination of these transports could be used in an integration solution, depending upon existing applications and other factors.

# Industry XML Connector

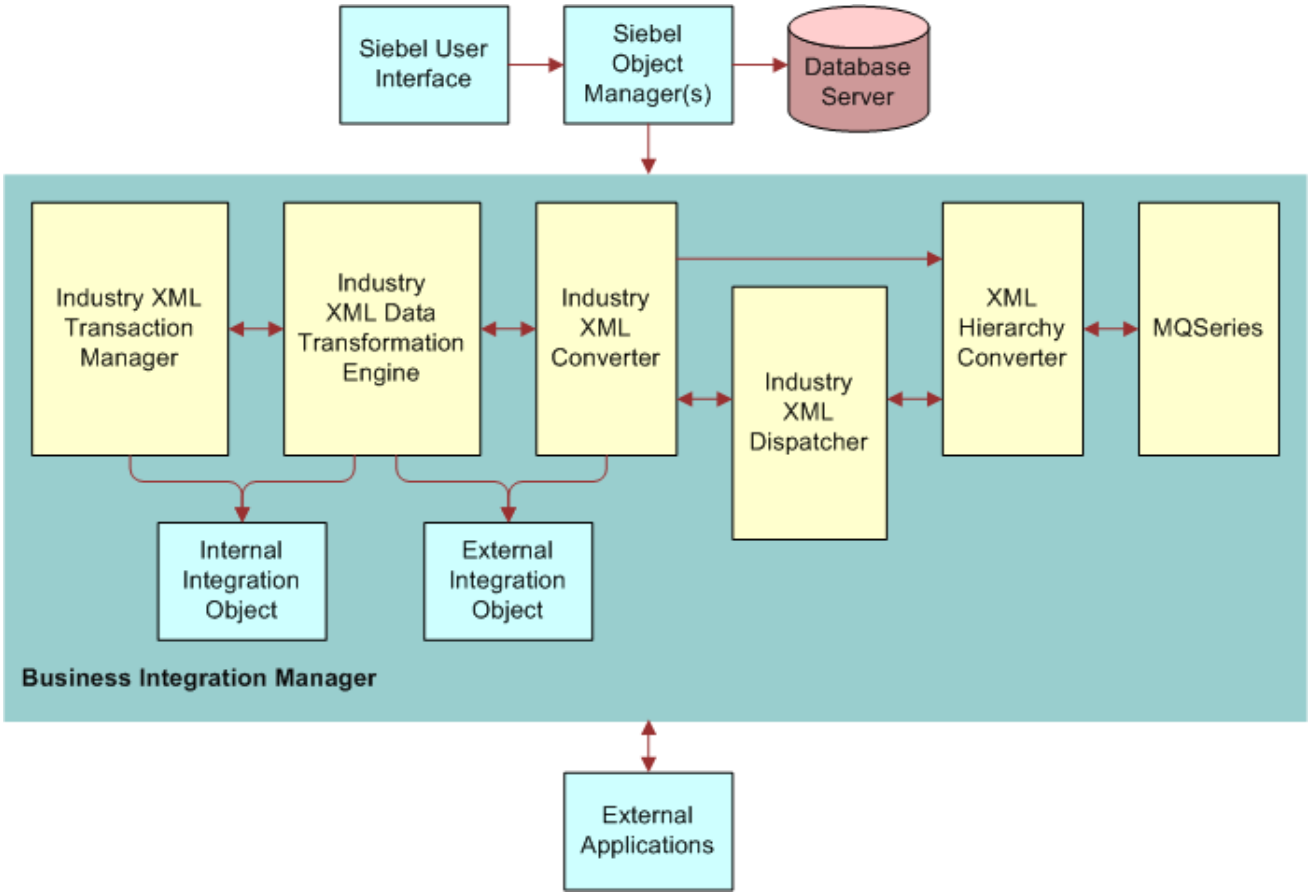
The Industry XML Connector is a configured set of components that allows Siebel applications to exchange data with external systems using XML messages. These components are called business services and can be extended to support multiple connectors. You can use the overwrite methods to extend the functionality of both industry XML connectors as well as Siebel Financial Service EAI infrastructure for enterprise data sharing and data synchronization for your integration implementation. By using the overwrite methods to extend the functionality, you obey the rules of data exchange set forth by the legacy or back-end systems, such as specific XML structure and envelope details, while using the common requirements for business processes and transactions. The following image on High-level architecture of Industry XML Converter illustrates architecture of the Industry XML Connector.

The Industry XML connector has four key components: Industry XML Transaction Manager, Industry XML Data Transformation Engine, Industry XML Converter, and Industry XML Dispatcher.

For more information about each of these components, see the following:

- *Industry XML Transaction Manager*
- *Industry XML Data Transformation Engine*
- *Industry XML Converter*
- *Industry XML Dispatcher*

**Note:** For details on the IFX XML integration solution, see *Siebel Financial Services Connector for IFX XML Guide* , for details on the ACORD XML integration solution, see Siebel Financial Services Connector for ACORD P&C and Surety Guide, and for details on implementing other industry XML integration solutions see *Industry XML Connector Factory APIs*




# Industry XML Transaction Manager

The Industry XML Transaction Manager is responsible for interfacing with the Siebel database. It retrieves data and executes database operations for financial services industry integration architecture. The Industry XML Transaction Manager can also extract query specifications entered by users in the memory cache, and run them through the connector modules to send a query message to external systems. Furthermore, it can undo operations if it receives an

error in the fault section of the response message indicating that the external systems failed to execute the specified operations.

The following table lists all the user properties for Industry XML Transaction Manager.

| Name | Value | Description |
|---|---|---|
| DispatcherMapName | IXMLDispMap | Transaction Manager uses this map to tag the Body information for other components. This value can also be set as a runtime input argument. |
| IgnoreSvcMethodArgs | true or false | This parameter allows runtime input arguments. |
| SaveInFileForRollback | <filename> | The filename to save current record for future rollback when rollback operation is desired. |
| SaveInMemForRollback | <filename> | The session key to set or look up the record cached in memory if rollback operation is desired. |
| XXX (Operation) | ServiceName/ MethodName/ Argument | Type of operation to use. For example, Name = IXMLOperation_ QUERY, Value= EAI Siebel Adapter/Query/PrimaryRowId;! SiebelMessage. |

The Industry XML Transaction Manager uses the prebuilt methods described in the following table to process inbound or outbound integration.

| Method | Default Display Name | Description |
|---|---|---|
| Execute | Execute Transaction | This method can be used for inbound or outbound purpose, as long as the integration object instance is provided. You should use the Execute Outbound method when Row Id is the only available input. |
| ExecuteOutbound | Execute Outbound | This method can only be used for outbound purpose to execute operation specified in input arguments. See the Method Arguments for ExecuteOutbound Method table. |
| ExecuteSave | Execute and Save | This method executes outbound operation specified in input arguments and also saves the transaction into memory or file. See the Method Arguments for ExecuteSave Method table. |

The Industry XML Transaction Manager business service uses combinations of user property, method, and method arguments to achieve different tasks. The following table lists all the method arguments available to use with The Industry XML Transaction Manager business service.

| Argument | Default Value | Description |
|---|---|---|
| OnlyIOI | false | This is only used for Inbound integration. The inbound message may contain header, body, and envelope portion. When Transaction |

| Argument | Default Value | Description |
|---|---|---|
| | | Manager takes the proper operation against the Siebel application, the integration object instances for response is generated as well. All information from the request message is dropped if this value is set to true. Therefore, the Industry XML Converter and Industry XML Data Transformation Engine do not need to deal with the overheads. If this is not set to true, then request information will still be carried over. |
| XMLHierarchy | N/A | Property set in internal integration object hierarchy. |
| IXMLMapPath | N/A | Store the key value for the dispatch map. Industry XML Transaction Manager will use it to look up the value, and attach necessary value into integration object instance. |
| PrimaryRowId | N/A | The PrimaryRowId of the integration object. |
| SiebelFINSOperationOut | N/A | The outbound operation Industry XML Transaction Manager takes through its user property. |
| SearchSpec | N/A | The Search Specification for query. |
| PlaceToSave | mem | PlaceToSave indicates whether the rollback instance is getting from “file" or “mem." Valid values are “file" or “mem." |
| RollbackInError | false | Indicates whether the transaction should be rolled back. Valid values are “true" or “false." |

Although these arguments are available to use with the Industry XML Transaction Manager, not all can be used with each method. The following tables list all the method arguments for each method.

| Argument | Display Name | Data Type | Type | Optional |
|---|---|---|---|---|
| OnlyIOI | Only produce Integration Object Instance | String | Input | Yes |
| XMLHierarchy | XML Property Set | Hierarchy | Input/Output | No |
| RollbackInError | Is Rollback in Error | String | Input | Yes |

| Argument | Display Name | Data Type | Type | Optional |
|---|---|---|---|---|
| IXMLMapPath | IXML Map Path | String | Input | No |
| PrimaryRowId | Primary Row Id | String | Input | Yes |

| Argument | Display Name | Data Type | Type | Optional |
|---|---|---|---|---|
| SiebelFINSOperationOut | Outbound Operation | String | Input | No |
| SearchSpec | Search Specification | String | Input | Yes |
| XMLHierarchy | XML Property Set | Hierarchy | Input/Output | No |

| Argument | Display Name | Data Type | Type | Optional |
|---|---|---|---|---|
| IXMLMapPath | IXML Map Path | String | Input | No |
| PrimaryRowId | Primary Row Id | String | Input | Yes |
| SiebelFINSOperationOut | Outbound Operation | String | Input | No |
| SearchSpec | Search Specification | String | Input | Yes |
| XMLHierarchy | XML Property Set | Hierarchy | Output | No |
| PlaceToSave | Place To Save | String | Input | Yes |

# Industry XML Data Transformation Engine

The Industry XML Data Transformation Engine transforms the hierarchy between external and internal data models that are represented by integration object instances. The Industry XML Data Transformation Engine is designed to handle heterogeneous messages to allow more than one business object scenario. It also takes extra input arguments specified in the data transformation map to indicate the possibility of removing empty value tags from the instances transformed.

The following table lists the specific methods for the industry XML Data Transformation Engine.

| Methods | Display Name | Description |
|---|---|---|
| ToExternal | Transform To External Hierarchy | Transforms Siebel internal integration object hierarchy into the industry XML external hierarchy. |
| ToInternal | Transform To Siebel Hierarchy | Transform industry XML external hierarchy into the Siebel internal integration object hierarchy. |

Industry XML Data Transformation Engine business service uses combinations of method, and method arguments to achieve different tasks. The following table lists all the method arguments available to use with Industry XML Data Transformation Engine business service.

| Argument | Default Value | Description |
|---|---|---|
| XMLHierarchy | N/A | Property set in internal or external integration object hierarchy. |
| <MapArgs> | N/A | See *Business Processes and Rules: Siebel Enterprise Application Integration* . |

Each method defined for the Industry XML Data Transformation Engine business service takes some input arguments and produces some output arguments as listed in the following tables.

| Name | Display Name | Data Type | Type | Optional |
|---|---|---|---|---|
| XMLHierrarchy | XML Property Set | Hierarchy | Input/Output | No |
| <MapArgs> | | String | Input | Yes |

| Name | Display Name | Data Type | Type | Optional |
|---|---|---|---|---|
| XMLHierrarchy | XML Property Set | Hierarchy | Input/Output | No |
| <MapArgs> | | String | Input | Yes |

# Industry XML Converter

The Industry XML Converter supports the generic syntax and rules requirements of the general industry XML standard. It converts the data from Siebel external integration object hierarchy to XML hierarchy that conforms to DTD specifications. For example, it generates certain authentication parameters for the XML message, and also generates and detects the fault sections in an XML message for extended converters to manipulate. The following table lists all the user properties for this converter.

| Name | Value | Description |
|---|---|---|
| EscapeNames | true or false | If set to true, the converter replaces illegal XML name characters with escape characters. Otherwise, it will ignore illegal XML name characters. The default value is true. |
| HierarchyFormat | XML Hierarchy | Format of the property set input or output. Internal use only. |
| XMLEnvIntObjectName | <integration object name> | Name of an integration object that defines the content and hierarchy for the envelope and header section of any industry XML standards. |

This service provides several ready-to-use-methods as described in the following table.

| Methods | Display Name | Description |
|---|---|---|
| PropSetToXML | PropSetToXML | Converts an integration object hierarchy to XML string. |
| PropSetToXMLPropSet | PropSetToXMLPropSet | Converts an integration object hierarchy to XML hierarchy. |
| XMLToPropSet | XMLToPropSet | Converts an XML string to an integration object hierarchy. |
| XMLPropSetToPropSet | XMLPropSetToPropSet | Converts an XML hierarchy to an integration object hierarchy. |

For each method defined the Industry XML Converter business service takes input arguments and produces output arguments. These arguments are described in the following table.

| Method Argument | Default | Description |
|---|---|---|
| XML Document | N/A | XML document streams. |
| Escape Names | true | Escape characters, invalid XML characters, to be removed or not. |
| Ignore Character Set Conversion Errors | false | If some characters cannot be represented in the destination character set, for example, local codepage, the errors can be ignored. |
| XML Character Encoding | N/A | XML character encoding to use in the output XML document. If encoding is blank or not supported, errors will be produced. |
| XML Header Text | N/A | Text to preappend to the XML document. |
| XML Hierarchy | N/A | Property Set in external integration object or XML hierarchy. |
| External Entity Directory | N/A | Location of an external entity file, such as a DTD file. |
| Validate External Entity | false | If true, the XML parser will validate the document according to the DTD specified in the <DocType> element. |
| Truncate Field Values | true | truncate field values. |
| Contains Inline Attachments | true | Indicates the message contains attachment documents. |

Each Industry XML Converter method uses groups of these method arguments to convert the data for the next service in the integration process. The following tables display the method arguments for each method.

| Name | Display Name | Data Type | Type | Optional |
|---|---|---|---|---|
| <Value> | XML Document | String | Output | Yes |
| EscapeName | Escape Name | String | Input | Yes |
| IgnoreCharSetConvErrors | Ignore Character String Set Conversion Errors | String | Input | Yes |
| XMLCharEncoding | XML Character Encoding | String | Input | Yes |
| XMLHeaderText | XML Header Text | String | Input | Yes |
| XMLHierarchy | XML Hierarchy | Hierarchy | Input | No |

| Name | Display Name | Data Type | Type | Optional |
|---|---|---|---|---|
| XMLHierarchy | XML Hierarchy | Hierarchy | Input/Output | No |

| Name | Display Name | Data Type | Type | Optional |
|---|---|---|---|---|
| <Value> | XML Document | String | Input | No |
| EscapeName | Escape Name | String | Input | Yes |
| ExternalEntityDirectory | External Entity Directory | String | Input | Yes |
| IgnoreCharSetConvErrors | Ignore Character String Set Conversion Errors | String | Input | Yes |
| ValidateExternalEntity | Validate External Entity | String | Input | Yes |
| XMLCharEncoding | XML Character Encoding | String | Output | Yes |
| XMLHierarchy | XML Hierarchy | Hierarchy | Output | Yes |

| Name | Default Display Name | Data Type | Type | Optional |
|---|---|---|---|---|
| ContainsInlineAttachments | Contains Inline Attachments | String | Input | Yes |
| TruncateFieldValues | Truncate Field Values | String | Input | Yes |
| XMLHierarchy | XML Hierarchy | Hierarchy | Input/Ouput | No |

# Industry XML Dispatcher

The Industry XML Dispatcher is responsible for receiving and dispatching inbound messages. It receives the message and determines the specific XML standards associated with the message. It also scans the message for commands that are supported and specified in its rule sets dispatcher map and associates each command with the corresponding integration objects and so on. The following table lists all the user properties for the Industry XML Dispatcher.

| Name | Value | Description |
|---|---|---|
| DispatcherMapName | <integration object name> | Name of an integration object that details the dispatching rules and syntax for the XML standard. This map is usually created along with all the other integration objects by the wizard. The default map name is “IXMLDispMap". |
| XMLEnvIntObjectName | <integration object name> | Name of an integration object that defines the content and hierarchy for the envelope and header section of any industry XML standards. |
| XMLFaultObject_0 | <path to fault section or element> | This allows the dispatcher to identify a fault section with the first token and further confirm it with the value of the second token if applicable. Extra fault objects can be added by incrementing the name with _1,_2, and so on. An example for the value for this user property name is XMLFaultObject_1 and value is //IOI/@cmdstatus;fail. |

The Industry XML Dispatcher business service provides ready-to-use methods described in the following table.

| Methods | Display Name | Description |
|---|---|---|
| DispatchMessage | Dispatch Message | Validates the incoming XML message. If the message conforms to the dispatching rules, integration object names and other necessary information will be attached. It also checks for the respective envelope, header, and fault section of the message and identifies them. |
| DispatchStdMessage | Dispatch Standard Message | Identifies the industry XML standard of the incoming XML message. It recognizes all XML standards configured in its input arguments. Value “Unknown" will be the output argument, if it does not recognize the XML standard. |

| Methods | Display Name | Description |
|---|---|---|
| | | |

For each method defined the Industry XML Dispatcher business service takes input arguments and produces output arguments. These arguments are described in the following tables.

| Argument | Default | Description |
|---|---|---|
| XML Hierarchy | N/A | Property Set in XML hierarchy. |
| <XML Standard> | N/A | An XML standard. For example, when configuring this input argument, the Input Argument is the XML standard and the Value is its envelope XML tag. |
| XML Standard | N/A | The XML standard that matches the incoming XML message's envelope tag. |

| Name | Display Name | Data Type | Type | Optional |
|---|---|---|---|---|
| XMLHierarchy | XML Hierarchy | Hierarchy | Input/Output | No |

| Name | Display Name | Data Type | Type | Optional |
|---|---|---|---|---|
| XMLHierarchy | XML Hierarchy | Hierarchy | Input | No |
| <XML Standard> | | String | Input | Yes |
| XMLStandard | XML Standard | String | Output | No |

# 4 Understanding Financial Services Integration Paradigms

## Understanding Financial Services Integration Paradigms

This chapter includes information about how to plan your integration.

## Overview

You need to integrate your Siebel applications with other external systems for the following reasons:

- You have invested in several legacy applications, back office systems, and eCommerce applications and would like to have one integrated view of all your systems.
- You want to use Siebel applications as your primary application and access all other applications using Siebel applications as the front-end system.
- You want to reflect changes made to your external systems in your Siebel applications and update your Siebel database with information from other applications in your enterprise.

Before you begin an integration project, identify your integration needs. You may have data in a variety of applications such as back-office solutions for Policy Administration, Claims, or accounting; e-commerce applications such as Web store fronts; and help desk applications.

There are several ways you can approach your integration, depending on your needs. You can connect Siebel applications and your back-office or e-commerce system using prebuilt Siebel Financial Services EAI connectors. You can connect Siebel applications and an external application, using Siebel Financial Services EAI custom-built connectors. You can integrate Siebel Financial Services applications with middleware solutions. And you can integrate Siebel Financial Services applications with external applications, using a third-party EAI product.

Siebel Financial Services EAI provides solutions for each element of the EAI planning process.

- Programmatic interfaces like COM and Siebel Java Beans provide a tightly-coupled integration for real time data exchange.
- Adapters like MQSeries and the HTTP adapters provide a loosely-coupled mechanism for real-time integration. This means that one system needs to be aware of implementation details of the other system's integration mechanisms. A message or a document is exchanged between these systems.
- For high volume batch integrations, Siebel EAI provides you with Batch Integration Manager (BIM) and Siebel Enterprise Integration Manager (EIM). BIM and EIM can replicate data between Siebel and other systems at the database layer, and both provide high performance for importing or exporting of large volume of data.
- Connectors, such as the Siebel Connector for IFX XML and other prebuilt connectors, provide loosely coupled integration using XML technologies. These connectors focus on existing industry XML standards.

Siebel Integration Designer provides the infrastructure that allows you to define real-time interfaces to external systems in a variety of ways. You can use the Siebel BIM to send and receive requests for data, requests for action,

and data synchronization. You can define integration workflow processes by sequencing integration requests without programming using Siebel Workflow Designer.

**Note:** For details on Siebel Workflow, see *Siebel Business Process Framework: Workflow Guide* .

# Design Decisions

Performing the correct analysis and making the correct decision is important for any software project but this is especially true for integration. Integrating disparate systems is a complex process because most customized applications that are created in an enterprise, and some commercially available applications, do not have a published API (Application Programming Interface). This makes seamless integration difficult, if not impossible. To add to this complexity, there are multiple infrastructures used by companies to “glue" systems together.

You need to carefully analyze the integration effort to understand your business needs and integration requirements. Before deciding to integrate your applications, you should perform a cost benefit analysis, review how you use the applications, and analyze the applications’ data structure. Some of the factors to be considered as part of this process include:

- Does the data need to be shared or synchronized?
- Does the data need to be transformed before it can be shared or synchronized?
- Is the requirement for data real-time or batch?

  **Note:** You will need more resources to build and maintain real-time integration than would be required by batch integration.

- How is the data presented to the user?
- How is the data requested?
- Where does the data come from?
- How is the data updated?
- What are the service-level or performance requirements?
- Is there an existing third-party product that can help?

Siebel Financial Service EAI offers different tool sets for different types of integration requirements. Siebel Financial Services EAI is a technology that allows for data sharing, data replication, and function passing across applications. Siebel Financial Services EAI includes the following technologies:

- Business Integration Manager
- Virtual Business Components
- Enterprise Integration Manager
- Business Services
- Business Object Interfaces
- ActiveX Plugins

There are several approaches to integrating the Siebel application with other applications based on the Business needs. Following are some of approaches which were considered:

Synchronize Siebel data with non-Siebel data. Synchronizing Siebel data with external applications can be achieved either through the Business Integration Manager, any of the Connectors, which Siebel ships as a part of the product, Enterprise Integration Manager (EIM), or any of the Object Interfaces.

Display non-Siebel data in Siebel applets. Displaying non-Siebel data within a Siebel application can be achieved through VBC or ActiveX Plugin.

Display Siebel data in another application. Displaying Siebel data within another application without the Siebel User Interface can be achieved by plugging Siebel data as an Active-X plugin, or through a Data Service of Object interface provided by Siebel applications.

Incorporate the Siebel User Interface (UI) into another application or incorporate non-Siebel UI within Siebel applications. Siebel provides an open interface which can be adapted using COM, Active X Controls, and Java Data Beans. You can use these technologies if an external application needs to have the look and feel of the Siebel application within its own user interface or if the Siebel application needs to incorporate the user interface of an external application.

Control the Siebel application from another application. You can use Object Interfaces such as COM Servers to have another application control Siebel applications.

Export Siebel data. To export data, you can use EIM or other Object Interfaces.

The following table lists the best Siebel solution for your integration problems.

| Integration Problem | Siebel Solution |
|---|---|
| High Volume Batch Interface | Enterprise Integration Manager |
| Accessing Data stored Outside of Siebel | Virtual Business Components (VBC) |
| Transactional Replication of Data | Business Integration Manager (BIM)<br><br>Siebel Financial Services Connectors |
| Interface to popular financial services back office applications using XML technologies | Siebel Financial Services Enterprise Application Integration Connectors (IFX XML, ACORD XML Connectors. Siebel Industry XML Connector, Siebel Industry XML Connector Factory APIs) |
| Interface to Popular Back-Office Applications (SAP, Oracle, PeopleSoft) | Siebel Connectors (SAP, Oracle, PeopleSoft Connectors) |
| External applications to control Siebel Application or Siebel application controlling external application | Object Interface such COM Servers (Automation and Data), Active X controls (Application and Data), and Java Data Beans |

To choose the appropriate and economical solution for your integration issue, you need to perform a step-by-step evaluation process for the following:

Transactions. List the business transactions that are executed within your Siebel implementation. These transactions should contain the information that is required for applications to integrate and work together. For example, if a company has a policy management system as one of its legacy systems but policy information initially is captured in the Siebel application, the transaction would be to add policy details in the policy management legacy system.

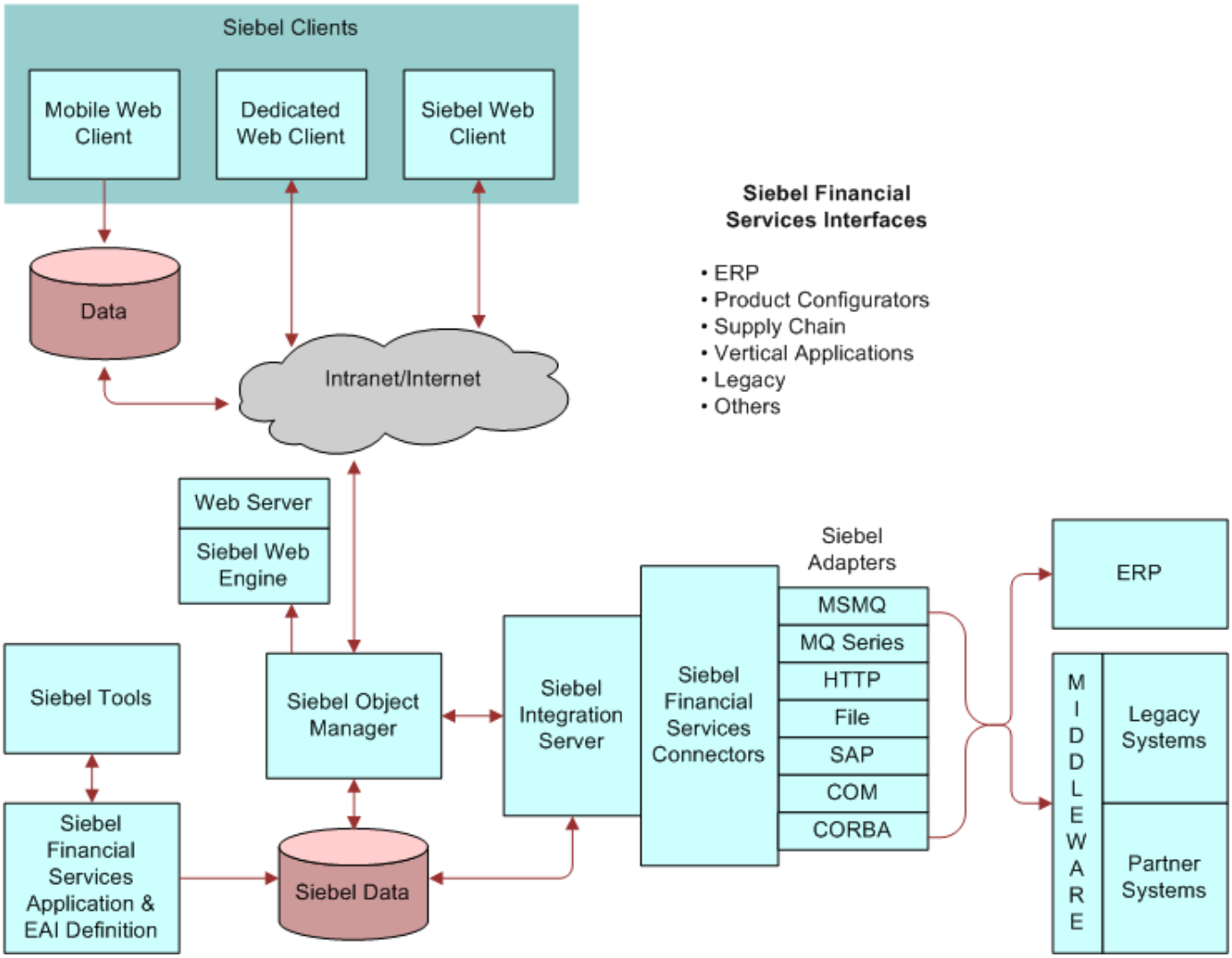
Applications. List the other applications you need to query or write to in order to complete the transactions. For example, in the previous case 'Policy Management System' would be the other application to integrate with the Siebel application.

Integration Points. Identify the points within the transaction flow at which the integration needs to take place (integration points). For example, populate a data warehouse with a nightly download of data. Consolidate integration points to reduce cost of ownership or verify open stock before showing the customer a product choice on a Web page. Add the transaction into the legacy system in real time as soon as a policy is added. Also identify what form of integration is required, real-time or batch, synchronous or asynchronous, tightly or loosely coupled.

Identify solutions. Identify the best solution for your integration need. There may be more than one solution in solving an integration problem. Consider the solution that is most optimal for your problem. You can use the table on Best Siebel Solutions for Integration Issues to analyze the most optimal solution for each of your integration points. For example, connector, middleware, VBC, or EIM.

Other Considerations. Determine whether you need to transform, share, or replicate your data. If you do not need to have the data in both systems but need to share it in real-time, consider VBC. Data sharing solves many of the problems associated with data replication. However, you need to consider data ownership contention and network traffic. Real-time access always carries a premium related to slow response time and the fact that it is expensive to maintain and build. Data sharing is appropriate for integration points that involve volatile or highly specialized source data such as credit card authorization, tax calculation, and so on. Other instances in which data sharing might be appropriate would be when you need to occasionally access large amounts of used data such as a credit card payment history. For details, see *Access Mechanisms*.

The following image illustrates available Siebel Financial Services interfaces.

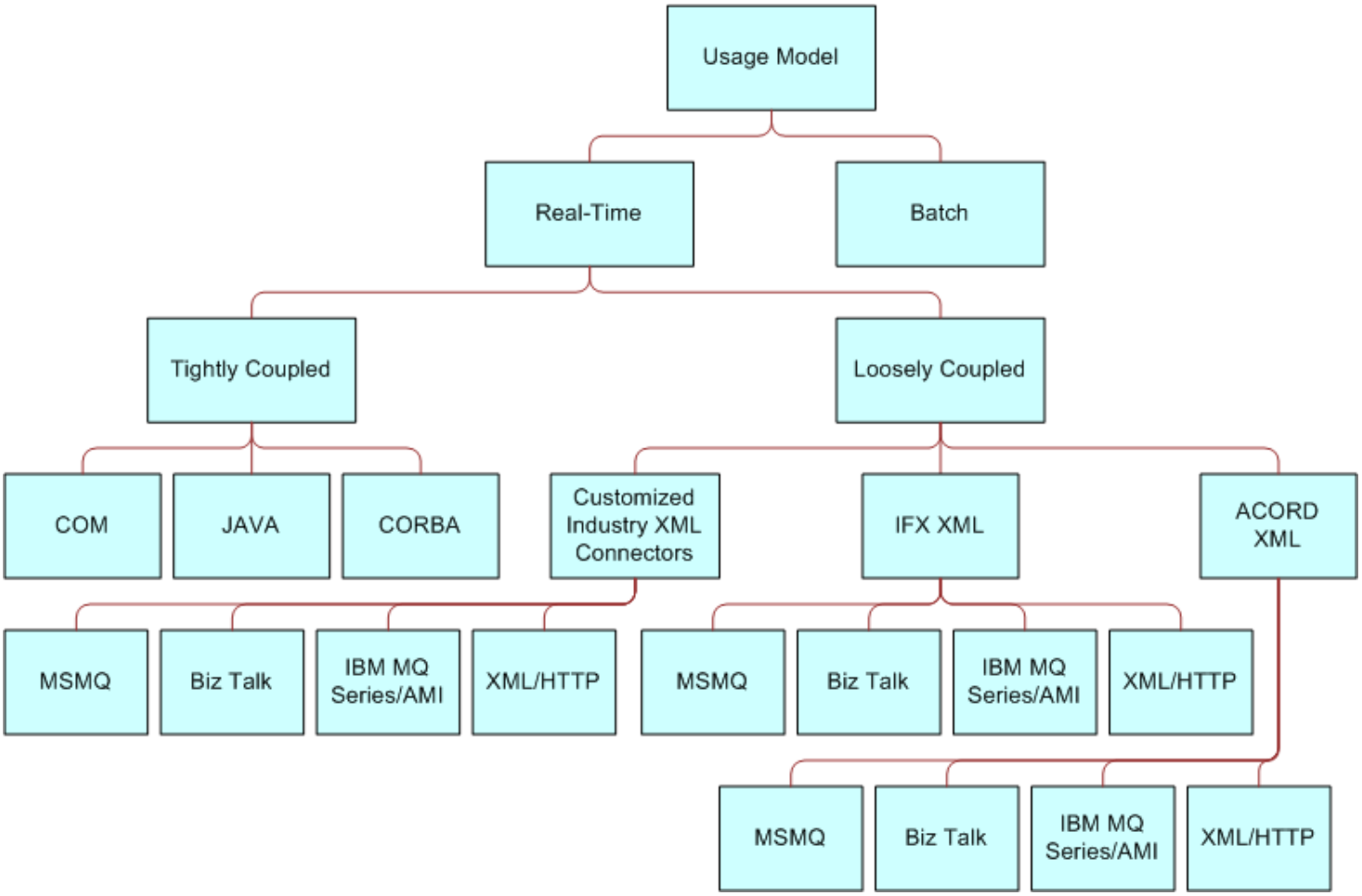



# Usage Models

When looking at your requirements, you should understand how you intend to use Siebel Financial Services EAI's integration capabilities. You need to determine how the integration will behave and under what circumstances. In other words, in what way are you planning to use this integration? This is referred to as the usage model.

There are two basic usage models, real time and batch, as shown in the Siebel Financial Services Interfaces image in *Design Decisions*. Batch mode has a single mode, and real time has two modes, either tightly-coupled or loosely-coupled. To help you determine the appropriate usage model, ask yourself the following questions.

- How do I plan to integrate Siebel Financial Services applications with other applications?
- Do I want to do it in real-time or in batch mode?
- If I want to do it in real-time, do I want the solution to be tightly-coupled or loosely-coupled?




# Real Time

With a real time model, you may choose to deploy Siebel Financial Services applications as a client or as a server. As a client, Siebel Financial Services applications access another program or initiate a transaction on an external system. Typical scenarios include:

- Sending an insurance policy created from Siebel Financial Services applications to a 'Policy Management' system and receiving the policy status in return.
- Accessing a mainframe banking application to retrieve customer bank deposits in real time and display it within the Siebel user interface.
- Sending claims to independent claims agents over corporate firewalls without compromising security.
- Accessing an external rating engine system to calculate a quote and then displaying the information in the Siebel system.

As a server, Siebel Financial Services applications are accessed by an external program to query, insert, update, or delete data without any initiation from the Siebel system. Typical scenarios include:

- An external EJB Web application accessing a Siebel Financial Services application to retrieve contact information.
- Call center software extracting customer service request information from a Siebel Financial Services application.

## Batch

Batch processing typically is used for periodic uploads or downloads of large amounts of information into the Siebel database. A typical batch scenario includes:

- Uploading a batch of product catalog and item information into a Siebel Financial Services application from an external system or downloading a batch of all opportunities from Siebel Financial Services applications to another database.
- Downloading all the new contact and accounts for that day from the Siebel application to an external Customer Information File (CIF).

# Access Mechanisms

When your business problem is accessing other applications from Siebel Financial Services applications or accessing Siebel Financial Services applications from other applications, you need to ask: Exactly what kind of data exchange do I require?

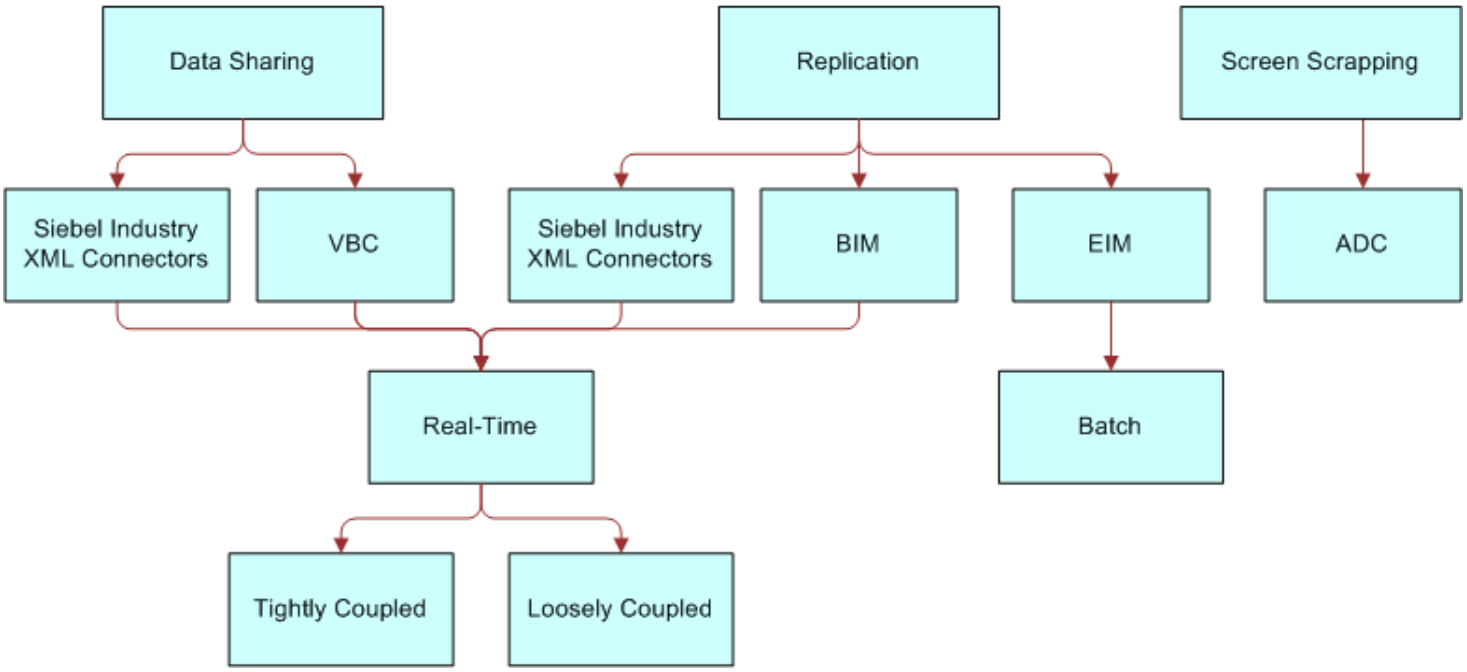
Data Sharing. You may just need to share data that exists outside of the Siebel system. For example you may have several legacy mainframe systems that you need to access with Siebel Financial Services to provide customers with comprehensive information on their insurance record. You might not want to replicate this information in the Siebel system, but may want to dynamically share this mainframe data as needed and display it within a Siebel Financial Services application. For data sharing, use VBC.

Replication. Replication of data is often useful. For example, you might want to integrate data in CIF (Customer Information File) with data in the Siebel system, keeping order status information in both systems. For mass replication of the order status information in Siebel Financial Services applications, use Siebel Business Integration Manager (BIM) and Siebel Enterprise Integration Manager (EIM).

Presentation Layer (UI) Integration. Integration at the presentation layer is useful in cases where you only need to look at the other application's user interface (UI) and the data does not need to be retrieved. You would capture just the "screen" from the external system using ActiveX Data Controls (ADCs) and represent it natively within a Siebel Financial Services application. This is the least expensive form of integrating two applications, however, it is also the least flexible.

Screen Scrapping. Screen scrapping is another approach you may want to consider. Screen scrapping is never a good long term solution mainly due to performance considerations, cost of ownership, and lack of change management functionality within the application. It is a temporary solution available from some Siebel partners. Screen scrapping can be a very powerful tactical weapon for a project that requires shorter development time and when the project's timeline needs to be separated from availability of legacy application or infrastructure resources.

These Siebel Financial Services solutions can be either real-time or batch. The following image shows the various access mechanism options and where they fit into the integration process.




# Application Connectors

Application connectors allow for the deployment of a point-to-point connection between Siebel Financial Services applications and other applications. These applications include IBM Websphere Composer, CBTF (Core Point Bank Teller Framework), and WSBCC (Web Sphere Business Component Composer). Oracle provides solutions for prepackaged connectors either directly or through partners and third-party developers. The Siebel Financial Services Solutions image in *Access Mechanisms* shows the prebuilt connectors that integrate with Siebel Financial Services applications.

# 5 Financial Services EAI Business Scenarios

## Financial Services EAI Business Scenarios

This chapter describes several common integration scenarios.

## Overview

Sometimes, the best way to determine your integration needs is to study examples of similar problems faced by other financial services organizations. This chapter presents three scenario categories, each with a number of permutations. The categories are:

- *Data Access and Replication*
- *Data Sharing Across the Enterprise*
- *Data Transformation*

This first step in getting started on an integration project is to identify your integration needs. Siebel Business Applications provide solutions for a variety of business requirements. They play a key role in the application environment in which you might use other applications, such as:

- Back-office solutions for claims processing, policy administration, or bank transaction management
- E-commerce applications as Web store fronts
- Mainframe-based applications that can store millions of customer transactions

As customer facing applications, Siebel Business applications need to interact with multiple external systems to provide an integrated view of customer information to the enterprise. Siebel Financial Services EAI provides the tools as well as prebuilt components that a company can use to allow this integration.

It is important to identify your needs prior to designing the integration. Your design choice could be based on the following:

- Data Transformation
  Data transformation between two systems is a key part of any integration project. You might want to implement this data transformation using the Siebel-provided data transformation functions or, if your organization has already standardized on a data transformation tool, you might choose to model the integration flow so that it uses the external tool for data transformation.
- Data Sharing
  Sometimes you might need only to view data from an external source within your Siebel applications, rather than needing replicated data. In this case, you might use a specific set of objects and tools designed explicitly for this purpose. For example, you might want to view credit card transaction details from within the Siebel applications. Your credit card transaction details might be stored in a mainframe system, whereas the Siebel applications maintain the contact information. When a customer requests credit card transaction details, you want to access the mainframe system and display the transaction details for that contact within your Siebel application, rather than replicating all the information within the Siebel application.
- Real-Time versus Batch

Another important issue to consider is what type of integration you need: real-time integration or a batch-mode process. In certain situations you might want to publish to the external system any new opportunities or contact information created or updated in the Siebel application as soon as the creation or update occurs. Likewise, you might want to exchange information in batch mode. For example, you might want to aggregate account information from all the customers financial institutions and load it into the Siebel application in batch mode.

# Common Integration Scenarios

The scenarios described in this chapter apply to many of the integration requirements you might face. Siebel Financial Services EAI allows you to accomplish integrations within the scope of these scenarios, but is not limited to the scenarios described here.

## Outbound Message to a File

In this scenario, you create a Siebel workflow process to create a unique report in a company standardized format such as XML. The file is to be sent to a central directory on a network server, where at midnight it is bundled automatically into a company standardized format for that day. These become part of a corporate knowledge base to go to management or head office.

## Round Trip Message from a File to Siebel and Back

In this scenario, you want all service requests entered from the company's EJB Web site to be entered as a service request in Siebel Service. You also want to pull an automatically generated service request number from Siebel Service, add it to the service request, and generate a file that is attached to a confirmation email message to the service requestor's email address.

## Round Trip Message from an External Application

In this scenario, whenever your back-office, for example ACORD based Policy Management, issues a new type of life product, it sends a message to your Siebel system to notify salespeople. If the salespeople like to issue such a policy, they can respond to the message and the response message will be generated and routed back to the back-office application indicating interest.

## Round Trip Message to an External Application

In this scenario, your salespeople want to be able to issue a new loan issued from Siebel Business Applications to companies credit rating systems such as Fair, Isaac's Liquid Credit System and receive a confirmation with a rating score generated by the Fair, Isaac's Liquid Credit System.

## Inbound Message from an External Application

In this scenario, whenever new auto policy information in the enterprise back-office (for example, ACORD-based Policy Management system) is updated, you want the system to send an update to your Siebel system in the form of an ACORD XML message containing the <PersAutoPolicyAddRq> message and have the policy information updated in the Siebel Asset business object.

## Outbound Message to an External Application

In this scenario, you want to update a subscribed policy modification made through your Siebel Business Application to the Policy Administration system. An outbound message containing the modification of customers' policy data is sent from the preconfigured business process in Siebel applications. The outbound message can be a standard ACORD XML message.

# Data Access and Replication

At times, you may need access to some specific data to accomplish a given task. Often it is not sufficient to bring everything to the same screen. Many business processes require access to particular data in order to accomplish a given task. This section describes the following four examples:

- *Accessing Customer History Information*
- *Exporting Contact Information*
- *Replicating a Company Catalog*
- *Updating Siebel Using Java Beans*

## Accessing Customer History Information

Financial Service institutions are typically very cautious in the way they treat customer data. However, they still need to provide access to data while replicating as few times as possible, if at all. Your Customer Service agents need to know the transaction history of your customers in order to serve them best. This information is typically stored in Customer Account Transaction Management systems on mainframe computers.

The following table details the requirements for integrating customer transaction history information into Siebel.

| Business Requirement | Technological Requirement | Oracle's EAI Solution |
|---|---|---|
| To access the Siebel database | Business service to query the integration object | Industry XML Transaction Manager<br>FINS ACORD XML Transaction Manager<br>FINS IFX XML Transaction Manager<br>Transaction Manager Factory Class |

| Business Requirement | Technological Requirement | Oracle's EAI Solution |
|---|---|---|
| To convert internal Siebel message format into different industry XML standards. | Siebel message-to-XML conversion technology | Industry XML Converter<br>FINS ACORD XML Converter<br>FINS IFX XML Converter<br>Converter Factory Class |
| To place the converted XML document in the destination queue | IBM MQSeries compatibility | EAI MQSeries AMI Transport<br>EAI MQSeries Server Transport |

Siebel Virtual Business Components (VBC) are built to handle just this business need. VBCs abstract external data and present it to the Siebel application as part of the Siebel data model. All the Siebel business logic and user interface components can process Virtual Business Components in exactly the same way that they handle regular Siebel Business Application components.

The following table displays the requirements for customer data access integration.

| Business Requirement | Technological Requirement | Oracle's EAI Solution |
|---|---|---|
| To provide access to the Customer Information Files (CIF) stored on a mainframe and display the information in the customer portal | Access to non-Siebel data without replication | Siebel Virtual Business Components (VBC) |

# Exporting Contact Information

Consider a sample business scenario where you want to send all changes to contact information in the Siebel database to an external mainframe system through the IBM MQ Series transport mechanism.

Your first step in this process would be to define and configure the message that needs to be exchanged. In this case, you want to exchange contact information. In the Siebel database, the Contact business object contains several business components and each business component has several fields. You do not want to send all of this information to the external system. You want to choose only a subset of this information. You also want to determine the XML standards the external systems implement and obtain the correct version of the XML standards schemas or DTDs.

The first task, therefore, is to use Siebel Tools to define the integration objects for the Contact contents in XML message you want to exchange. Siebel Tools provides an Integration Object Builder that walks you through the process of defining your integration objects. Once you define the message, you define what you want to do with this message.

You can now use the Siebel Workflow Process Manager to model this integration message flow to send the message to the external system. An integration message flow is a sequence of business services that are connected. For information on configuring messages, see:

- *Siebel Financial Services EAI Architecture*
- *Industry XML Connector Factory APIs*
- Siebel Financial Services Connector for ACORD P&C and Surety Guide

- *Siebel Financial Services Connector for IFX XML Guide*

## Replicating a Company Catalog

In this scenario, you need to provide your sales people access to a price list while they are on the road. So, you need to import the price list information into Siebel so that Oracle's Siebel Remote can take care of the mobile replication. The Enterprise Integration Manager (EIM) can take care of this task for you.

The following table details the requirements for importing a company's catalog for mobile replication.

| Business Requirement | Technological Requirement | Oracle's EAI Solution |
|---|---|---|
| To import the company's product catalog | High-volume batch replications | Siebel Enterprise Integration Manager (EIM) |

## Updating Siebel Using Java Beans

In this scenario, you have a Customer Service Web application written using J2EE Server Pages (JSPs). To pass this data into Siebel so that account records can be updated, you invoke the Siebel Object Interface, Siebel Java Data Beans. Now your Customer Service representatives can use the same information from the Java Server Pages while serving the customer.

The following table details the requirements for updating Siebel using JSP technology.

| Business Requirement | Technological Requirement | Oracle's EAI Solution |
|---|---|---|
| To update Siebel account data from a J2EE Server Page (JSP) enabled application | Programmatic interaction | Siebel Java Data Beans Object Interfaces |

# Data Sharing Across the Enterprise

Business processes often need to cross departmental or even enterprise boundaries. Because different organizations may be using different applications, you need a way to make sure one application can hand off the execution of a business process to the next application. This section describes three examples:

- B2B Integration
- Legacy Data Integration
- ERP Integration

# B2B Integration

To allow business-to-business transactions to take place over the Internet, Siebel Financial Services EAI provides HTTP adapter and the Business Integration Manager, which are used to model and execute business processes. In case the organization supports IFX XML or ACORD XML standards for B2B integration, Siebel provides connectors that work with the HTTP adapter. Furthermore, organizations may implement the Siebel Industry XML Connector Factory APIs for building customized XML connectors. The following table details the requirements and the solution for a successful B2B integration.

| Business Requirement | Technological Requirement | Oracle's EAI Solution |
|---|---|---|
| To create a business-to-business integration to send an order directly to a partner over the Internet for fulfillment | XML messaging over HTTP | • Siebel Workflow Process Manager<br>• EAI HTTP Transport adapter<br>• Siebel Connectors for different industry XML standards<br>• Siebel Industry XML Connector Factory APIs<br>• Business Integration Manager (BIM) |

# Legacy Data Integration

On the Internet, HTTP is the standard protocol for exchanging data. In a mainframe environment, IBM MQ Series is the most common application messaging transport. Therefore, the HTTP protocol can be replaced with MQ Series and still carry out the transaction. In case the organization supports IFX XML or ACORD XML standards for legacy data integration, Siebel provides Connectors that work with the HTTP adapter as well as IBM's WSBCC and MQSFSE product suite. In addition, they can be customized to support different message sets and packages to suit your business needs. The following table details the requirements and the solution for a successful integration of legacy data existing on a mainframe.

| Business Requirement | Technological Requirement | Oracle's EAI Solution |
|---|---|---|
| To create an order against a mainframe system | Cross-application business process management | • Siebel Workflow Process Manager<br>• EAI MQ Series transport adapter<br>• Siebel Connectors for different industry XML standards<br>• Siebel Industry XML Connector Factory APIs |

# ERP Integration

In this scenario, the main purpose is the correct and timely fulfillment of an order, but the back-end is now a popular ERP application. Siebel has developed prebuilt solutions that interface with SAP R/3, PeopleSoft, and with Oracle applications (both for the character-based 10.7 release and the GUI and web client release 11i). These applications are

set up to exchange order information, customer information, and so on, out-of-the-box. Additionally, they can be customized as needed to exchange as much or as little information from these external applications as your situation requires.

The following table details the requirements and the solution for a successful integration of ERP data.

| Business Requirement | Technological Requirement | Oracle's EAI Solution |
|---|---|---|
| To create an order against SAP R/3 | Integration with popular applications | Siebel Enterprise Application Integration Connector for SAP |

# Data Transformation

When performing the data transformation within the Siebel environment, you use two types of integration objects.

- Internal integration objects mirror the data structure of a Siebel business object and contain a subset of business components and fields.
- External integration objects mirror the data structure of an external system. You define integration objects of both types in Siebel Tools.

The Industry XML Transformation Engine performs the same function whether the data transformation is performed inside or outside of the Siebel environment. The Industry XML Converter and the transports also perform the same function, whether used within the Siebel environment or outside the Siebel environment. The only difference is that when used outside the Siebel environment, these business services perform operations on external integration object instances as opposed to performing operations on internal integration object instances when used within the Siebel environment.

Each of the Siebel Connectors for industry XML standards (IFX XML and ACORD XML) has a slightly different data transformation process that is specific to the industry that uses it. For more information about the data transformation process and connector configuration procedures, see *Siebel Financial Services EAI Architecture* Siebel Financial Services Connector for ACORD P&C and Surety Guide, and *Siebel Financial Services Connector for IFX XML Guide* .

# 6 Predefined Financial Services EAI Business Services

## Predefined Financial Services EAI Business Services

This chapter describes the predefined Siebel Financial Services EAI business services.

## Predefined Financial Services EAI Business Services

Siebel Financial Services Enterprise Applications provide a number of business services in addition to business services provided by Siebel Business Applications. These services do not require any modification, but do require that you choose and configure them to suit your business requirements.

**Note:** For information on using business services, see *Overview: Siebel Enterprise Application Integration* , and for a list of business services provided by Siebel Business Applications, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

The following table describes the predefined Siebel Financial Services EAI business services.

| Business Service | Class | Description |
|---|---|---|
| FINS Industry XML Transaction Manager | CSSFAIXMLTransMgrService | Industry XML Transaction Manager Service. |
| FINS Industry XML Data Transformation Engine | CSSFAIXMLDTEService | Industry XML Data Transformation Engine Service. |
| FINS Industry XML Converter | CSSFAIXMLCnvService | Industry XML Converter Service. |
| FINS Industry XML Dispatcher | CSSFAIXMLDispService | Industry XML Dispatcher Service. |
| FINS Industry XML Query Service | CSSFAEAIUtilService | The business service to query the XML instance using XPath. |
| FINS Industry XML BC Facility Service | CSSFAIXMLBCFService | The business service to get the search specification. |
| FINS ACORD XML Transaction Manager | CSSFAACORDIXMLTransMgrService | ACORD XML Transaction Manager Service. |

| Business Service | Class | Description |
|---|---|---|
| FINS ACORD XML DTE | CSSFAIXMLDTEService | ACORD XML Data Transformation Engine Service. |
| FINS ACORD XML Converter | CSSFAACORDBASEIXMLCnvService | ACORD XML Converter Service. |
| FINS ACORD XML Dispatcher | CSSFAIXMLDispService | ACORD XML Dispatcher Service. |
| FINS ACORD Wizard Service | CSSFAACORDUIService | The business service to create ACORD XML external and Siebel internal integration objects. |
| FINS IFX XML Transaction Manager | CSSFAIFXIXMLTransMgrService | IFX XML Transaction Manager Service. |
| FINS IFX XML DTE | CSSFAIXMLDTEService | IFX XML Data Transformation Engine Service. |
| FINS IFX XML Converter | CSSFAIFXIXMLCnvService | IFX XML Converter Service. |
| FINS IFX XML Dispatcher | CSSFAIXMLDispService | IFX XML Dispatcher Service. |
| FINS IFX Wizard Service | CSSFAIFXUIService | The business service to create IFX XML external and Siebel internal integration objects. |
| FINS OFX Wizard Service | CSSFAOFXUIService | The business service to create OFX XML external and Siebel internal integration objects. |

# 7 Industry XML Connector Factory APIs

## Industry XML Connector Factory APIs

This chapter describes the Industry XML Factory Classes and their methods.

## Industry XML Connector Factory Classes

The Industry XML Connector Factory Classes are developed based on the Siebel Financial Services EAI infrastructure aimed at supporting variety XML Standards extension for the Industry XML Connector described in *Siebel Financial Services EAI Architecture* The factory classes are built to leverage the functionality of Industry XML Connector while giving customers the opportunity to customize generic connector business services through configuration and programming scripts. In most cases, scripts are needed for presentation purpose, but in rare cases they may be used to add or enhance the connector functionality.

### To set up customized XML Connector

1. Configure the customized connector by configuring the four business services based on the factory classes, configure methods, and method arguments for each business service which should be identical to Industry XML Connector business services.

   **Note:** For a list of methods and method arguments available for the four business services, see *Siebel Financial Services EAI Architecture*

2. Customize the behavior of the connector through the user properties of the four business services configured in Step 1.

   **Note:** For a list of available user properties for the four business services, see *Siebel Financial Services EAI Architecture*

3. Overwrite any extension method from the factory classes to fulfill the different syntax and rules of a desired XML standard when applicable.

   **Note:** For details, see *Industry XML Connector Factory Classes Methods* and *Siebel Connector for Credit Card Application XML*.

4. Generate all the necessary integration objects and deliver the changes to the integration branch.
5. Configure the data transformation maps to map the internal and external integration objects.
6. Configure the business logic and flow using workflow and invoke these four customized business services in the workflow steps.

# Industry XML Connector Factory Classes Methods

The Industry XML Connector Factory consists of four main classes:

- Transaction Manager
- Data Transformation Engine
- Converter
- Dispatcher

**Note:** For a description of functionality of each class, see *Industry XML Connector*.

Each class exposes a number of methods to fulfill customization and extension. The collection of these methods made up the Industry XML Connector Factory APIs. Using the overwrite methods, you can use the functionality of both Industry XML Connector as well as Siebel Financial Services EAI infrastructure for enterprise data sharing and data synchronization in your integration implementation while obeying the rules of data exchange set forth by the legacy or back-end systems such as specific XML structure and envelope details.

**Note:** Only methods absolutely needed should be overwritten to modify the generic behavior. A list of available methods for each module is provided later in This chapter.

## Transaction Manager Factory Class Methods

The Transaction Manager Factory Class (CSSFAELTransMgrService) provides the methods described in the following table that you can overwrite.

| Method | Description |
|---|---|
| PreProcess | You can use this method to modify or reserve information within the integration object instance. |
| PostProcess | You can use this method to modify or restore stored information into the integration object instance. |

## Data Transformation Engine Factory Class Methods

The Data Transformation Engine Factory Class (CSSFAELDTEService) provides the methods described in the following table that you can overwrite.

| Method | Description |
|---|---|
| PreProcess | Allows application developers to modify or reserve information within the integration object instance. |

| Method | Description |
|---|---|
| PostProcess | Allows application developer to modify or restore stored information into the integration object instance. |
| ProcessMessage | Allows application developer to process information within the root tag of message. |
| ProcessHeader | Allows application developer to process information within the header portion of message. |

# Converter Factory Class Methods

The Converter Factory Class (CSSFAELCnvService) provides the methods described in the following table that you can overwrite.

| Method | Description |
|---|---|
| GenerateEnvelope | Allows application developer to prepare the envelope section of the XML message to be sent. |
| GenerateHeader | Allows application developer to prepare the header section of the XML message to be sent. |
| GenerateError | Allows application developer to generate the fault section of the XML message to be responded to. |
| GeneratePI | Allows application developer to prepare the processing instruction section of the XML message to be sent. |
| ProcessEnv | Allows application developer to process the envelope section of the XML message received. |
| ProcessHdr | Allows application developer to process the header section of the XML message received. |
| ProcessError | Allows application developer to process the fault section of the XML message received. |
| ProcessPInst | Allows application developer to process the processing instruction section of the XML message received. |
| PreProcessBody | Allows special processing of the body section of the XML message. |
| PostProcessBody | Allows special processing of the body section of the XML message after conversion is done. |
| StartProcess | Allows special processing of the body section of the XML message for each command. |
| EndProcess | Allows special processing of the body section of the XML message for each command after conversion is done. |

## Dispatcher Factory Class Methods

The Dispatcher Factory Class (CSSFAELDispService) provides the methods that you can overwrite. These methods are described in the following table.

| Method | Description |
|---|---|
| DispatchEnvelope | Allows application developer to identify the envelope section of the message, tag it, and insert other information such as its matching integration object involved. |
| DispatchHeader | Allows application developer to identify the header section of the message, tag it, and insert other information such as its matching integration object name. |

# Industry XML Connector Factory Method Arguments

All extension methods for the factory classes are based on the Siebel Property Set data structure, with no exception. The following table lists the method arguments.

| Argument | Description |
|---|---|
| XMLHierarchy | Input Property Set |
| XMLHierarchy | Output Property Set |

# Siebel Connector for Credit Card Application XML

The following example shows the extension connector that implements the Industry XML classes through the sets of APIs. It shows how each of the steps is carried out to achieve supporting the credit card application services.

## Sample Result

```
- <CRM>
 - <SignonRq>
 - <UsrLoginSegment>
 <LoginUserId>UserId</LoginUserId>
 <LoginCurPassword>PASSWD</LoginCurPassword>
 <LoginNewPassword />
 </UsrLoginSegment>
 </SignonRq>
- <CreditCardSvcRq>
 - <CardAppAddRq>
```

```
- <AccountData>
<UserId>1-DLN</UserId>
<CompanyCd>1-109</CompanyCd>
<DivisionCd>1-109</DivisionCd>
<ServCtrCd>1-109</ServCtrCd>
<BranchCode>1-16UU</BranchCode>
</AccountData>
- <ApplicationInfo>
<ApplicantId>AAA CreditCard Join</ApplicantId>
<PrioritySortCode />
<ApplSourceCode />
<WorklistModeFlag>03 - Qualification</WorklistModeFlag>
<ApplicationClass>Credit Card Application</ApplicationClass>
</ApplicationInfo>
- <CardholderInfo>
<FirstName>Enid</FirstName>
<Mi>J</Mi>
<LastName>Alberts</LastName>
<City>San Jose</City>
<State>CA</State>
<ZipCode>95110</ZipCode>
</CardholderInfo>
</CardAppAddRq>
</CreditCardSvcRq>
</CRM>
```

## To implement the industry XML classes through the sets of APIs

1. Configure the following four business services to support all the methods and method arguments in Industry XML Connector:
   - FINS EL XML Transaction Manager
   - FINS EL XML Data Transformation Engine
   - FINS EL XML Converter
   - FINS EL XML Dispatcher

   For example, the FINS EL XML Data Transformation Engine business services has Execute, ExecuteOutbound, and ExecuteSave methods.

   > **Note:** The “FINS Connector Factory" project, in the Sample database, includes all the business services configurations. You can access the project through Siebel Tools.

2. Configure user properties for the four business services configured in Step 1.

   > **Note:** The “FINS Connector Factory" project, in the Sample database, includes all the user properties configurations. You can access the project through Siebel Tools.

3. Overwrite a few (necessary) Industry XML APIs to modify the presentation logic.

   The following example shows the customized converter over the functions to generate message header and preprocess the body section. In the GenerateHeader function, the application calls the CreateSignOn function to generate the login information portion of the XML document. In the PreProcessBody function, the service aggregate `<CreditCardSvcRq>` is added to embed the `<CardAppAddRq>` message aggregate. Your scenario may be different.

**Note:** The common usage of MethodName in standard scripting is a publicly available business service method such as PropSetToXMLPropSet. However, in this example, the MethodName does not correspond to the business service method when implementing the factory classes. Instead, it refers to the methods in the Industry XML APIs.

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{
 var retVal = CancelOperation;
 if (MethodName == "GenerateHeader") {
 Func_GenerateHeader(Inputs, Outputs);
 }
 else if (MethodName == "PreProcessBody") {
 Func_PreProcessBody(Inputs, Outputs);
 }
 else {
 retVal = ContinueOperation;
 }
 return (retVal);
}
// Overwrite GenerateHeader method to handle SignonRq message
function Func_GenerateHeader (Inputs, Outputs)
{
 var child;
 // Enumerate the first layer header definition
 for (var i = 0; i < Outputs.GetChildCount(); i++)
 {
 child = Outputs.GetChild(i);

 if ( child.GetType() == "SignonRq")
 {
 // Create SignOn Segment
 Func_CreateSignOn(child);
 }
 else if (child.GetType() != "xxxSvcRq")
 {
 // Remove other header definitions
 Outputs.RemoveChild(i--);
 }
 }
}
// Overwrite PreProcessBody method to add service aggregate
function Func_PreProcessBody (Inputs, Outputs)
{
 var serviceName;

 if (Inputs.PropertyExists("SiebelFINSObjType"))
 {
 Func_CopyPropSet(Inputs, Outputs);
 }
 else if (Inputs.PropertyExists("IFX Service Name"))
 {
 serviceName = Inputs.GetProperty("IFX Service Name");
 // add CreditCardSvcRq aggregate
 Outputs.SetType(serviceName);

 // copy CardAppAddRq aggregate
 Outputs.AddChild(Inputs.Copy());
 }
 else
 {
 Func_CopyPropSet(Inputs, Outputs);
 }
}
```

```
// Sign on utility to access user name and password
function Func_CreateSignOn(Inputs)
{
 var child;
 var grdChild;
 // Enumerate the first layer header definition
 for (var i=0; i < Inputs.GetChildCount(); i++)
 {
 child = Inputs.GetChild(i);
 if ( child.GetType() == "UsrLoginSegment")
 {
 for (var j=0; j < child.GetChildCount(); j++)
 {
 grdChild = child.GetChild(j);
 if (grdChild.GetType() == "LoginUserId")
 {
 grdChild.SetValue("UserId");
 }
 if (grdChild.GetType() == "LoginCurPassword")
 {
 grdChild.SetValue("PASSWD");
 }
 }
 }
 else
 {
 // Remove other header definitions
 Inputs.RemoveChild(i--);
 }
 }
}
// Generic copy property set utility
function Func_CopyPropSet( Inputs, Outputs)
{
 var propName = "";
 var propVal = "";
 var child;
 Outputs.SetType(Inputs.GetType());
 Outputs.SetValue(Inputs.GetValue());
 propName = Inputs.GetFirstProperty();
 while (propName != "")
 {
 propVal = Inputs.GetProperty(propName);
 Outputs.SetProperty(propName, propVal);
 propName = Inputs.GetNextProperty();
 }
 for (var i = 0; i < Inputs.GetChildCount(); i++)
 {
 child = Inputs.GetChild(i);
 Outputs.AddChild(child.Copy());
 }
}
```

4. Configure integration objects as required.

   Some integration objects have been configured for this example. The “FINS Connector Factory" project in Oracle’s Siebel Tools includes all the integration objects.
5. Configure data transformation maps as required.

   Two data transformation maps have been configured for this example. You can list them in Oracle’s Siebel Financial Services application with a query specification of `"Credit Card*Add*"` in the Comments field of the Integration Object Map screen.
6. Configure workflows as required.

Two workflows have been configured for this example: "Credit Card Application Inbound Workflow" and “Credit Card Application Outbound Workflow."