SIEBEL eBusiness 7

SIEBEL UNIVERSAL CUSTOMER MASTER APPLICATIONS REFERENCE FOR INDUSTRY APPLICATIONS

VERSION 7.5, REV. A

12-EECGVH

JANUARY 2003

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404

Contents

Introduction

Chapter 1. About Siebel UCM Applications

Chapter 2. Siebel UCM Applications Scenarios

Chapter 3. Architecture and Framework

Chapter 4. Siebel UCM Applications Administration

Chapter 5. Siebel UCM Applications Messages

Appendix A. Additional License Restrictions

Introduction

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

Business Analysts Persons responsible for analyzing application integration challenges and planning integration solutions at an enterprise.

Database Administrators Persons responsible for administering the database system, including data loading, system monitoring, backup and recovery, space allocation and sizing, and user account management.

Siebel Application Administrators Persons responsible for planning, setting up, and maintaining Siebel eBusiness Applications.

Siebel Application Developers Persons responsible for the planning, implementing, and configuring of Siebel eBusiness Applications, possibly adding new functionality.

Siebel Integration Developers Persons responsible for analyzing a business situation or using the analysis of a Business Analyst to build the integration solution at an enterprise for Siebel eBusiness Applications.

Siebel System Administrators Persons responsible for the whole system, including installing, maintaining, and upgrading Siebel eBusiness Applications.

System Integrators Persons responsible for analyzing a business situation or using the analysis of a Business Analyst to build the integration solution at an enterprise for specific applications and to develop custom solutions.

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

How This Guide Is Organized

This guide is organized as follows:

Chapter 1	Provides an overview of Siebel Universal Customer Master Applications (Siebel UCM Applications).
Chapter 2	Provides a business scenario and various deployment options for Siebel UCM Applications.
Chapter 3	Explains in further detail the framework and architecture of Siebel UCM Applications and the interactions between Siebel UCM Applications and other Siebel eBusiness Applications or back-office systems.
Chapter 4	Details procedures necessary to administer Siebel UCM Applications.
Chapter 5	Provides additional information on messages used with Siebel UCM Applications.
Appendix A	Contains additional license restrictions for Siebel UCM Applications.

Siebel UCM Applications interact with many other standard Siebel products, especially Siebel eBusiness Application Integration (Siebel eAI). Reviewing documentation on Siebel eAI and other associated Siebel products is recommended. Please see "Siebel eBusiness Applications with Siebel UCM Applications" on page 13 for further information about this documentation.

NOTE: Chapters 3, 4, and 5 use former naming conventions for the Siebel UCM Applications product when referencing hard-coded product screens and views, business services, and interface messages. References to CIF (Customer Information File) and FINS CIF (Financial Services Customer Information File) are valid for the Siebel UCM Applications for Industry Applications product.

Revision History

Siebel Universal Customer Master Applications Reference for Industry Applications, Version 7.5, Rev. A

This guide has been retitled from *Siebel Information File Applications Reference for Industry Applications*, Version 7.5.

January 2003 Bookshelf

Table 1. Changes Made in Rev. A for January 2003 Bookshelf

Topic	Revision
Chapter 1, "About Siebel UCM Applications"	Updated text and references.
"Siebel eBusiness Applications with Siebel UCM Applications"	Added information on D&B Integration product.
"Business Solution"	Added two bullet points.
"Siebel Universal Customer Master Application Messages"	Added list and details of Siebel Customer Master Application messages.

Additional Changes

- Changed the name of the product from Siebel Information File Applications to Siebel Universal Customer Master Applications (Siebel UCM Applications) throughout the book.

About Siebel UCM Applications 1

Siebel Universal Customer Master Applications (Siebel UCM Applications) provide a Siebel platform that functions as the master file for an organization's enterprise-wide customer information. It consists of the following master file modules:

- Siebel Universal Customer Master
- Siebel Advanced Customer Profile
- Siebel Product Master
- Siebel Marketing Master
- Siebel Sales Master
- Siebel Service Master

The Siebel Universal Customer Master is the base module on which all other master file modules build. These modules add extended enterprise-wide information capabilities. The Siebel Universal Customer Master module consists of a set of Siebel data model tables that store customer data; the other modules extend this concept to hold additional marketing, sales, and service data.For information about Siebel UCM Applications, see "Siebel UCM Applications Modules" on page 11. Siebel UCM Applications act primarily as an information database and are not intended to function as an end-user application.

When Siebel UCM Applications are used as the master database of an organization's data, one or more of the modules—through proprietary and third-party software—interact with back-office systems and Siebel eBusiness Application deployments to provide a unified data-set across an organization's multiple channels, lines of business, and applications.

In addition to the data model mentioned previously, Siebel UCM Applications include a data manager, an object manager, and a limited interface used for administrative tasks. A standard Siebel eBusiness Application user interface is not included with Siebel UCM Applications.

Siebel UCM Applications can be implemented in different scenarios with or without standard Siebel eBusiness Applications. Siebel UCM Applications can also interact with other third-party back-office systems through any of the following means:

- By invoking Universal Application Network integration business processes currently available with UAN Customer Lifecycle Management. See *Integration Business Process Guide: Universal Application Network Volume 3* for further information.
- By directly invoking the Application Services Interfaces, which can be published as Web Services, Java Data Beans, and so on. See *Application Services Interface Reference* for further information.
- By using any previously available Siebel integration technologies such as Siebel eAI, EIM (Enterprise Integration Manager), MQ Series Adapter, COM, CORBA, and so on.

Also see "Framework and Interactions of Siebel UCM Applications" on page 26 for further information.

Siebel UCM Applications Modules

Siebel UCM Applications include the base module, Siebel Universal Customer Master, and five additional modules comprising various subsets of enterprise-wide customer related data. See module list below:

- "Siebel Universal Customer Master"
- "Siebel Advanced Customer Profile"
- "Siebel Product Master" on page 12
- "Siebel Marketing Master" on page 12
- "Siebel Sales Master" on page 12
- "Siebel Service Master" on page 12

Siebel Universal Customer Master

The Siebel Universal Customer Master is a Siebel platform configured to store a clear and unified profile for enterprise customers, partners, and prospects. Traditional customer data such as Accounts, Contacts, Households, Partner, and Agent data is included as well as customer relationship information, address information, and asset information. The Siebel Universal Customer Master interacts within an enterprise architecture by integrating with key back-office systems to act as the master record for the customer-specific subset of an organization's data.

The Siebel Universal Customer Master is based on the Siebel party data model. This model uses a single-party entity to represent organizations, positions, user lists, contacts, and employees and is implemented with Siebel extension tables. Each record of the party data model table can have complex hierarchical relationships with other parties.

Siebel Advanced Customer Profile

The Siebel Advanced Customer Profile module is a data model extension that stores additional customer information. This extension creates an extended profiling view of an organization’s customers. The key entities included with this module are activities, notes, agreements, entitlements, billing accounts, demographic information, business profiles, customer preferences, credit profiles, activities history, and analytical and segmentation information.

Siebel Product Master

The Siebel Product Master module is a data model extension that stores an organization’s product information. This extension creates a common and consistent definition of an organization’s products. The key entities included with this module are product line management, product configuration, complex products, and price list.

Siebel Marketing Master

The Siebel Marketing Master module is a data model extension that stores an organization’s marketing information. This extension creates a central storage location for marketing efforts across an organization’s multiple channels. The key entities included with this module are campaigns, offers, and responses.

Siebel Sales Master

The Siebel Sales Master module is a data model extension that stores an organization’s sales information. This extension creates a central storage location for sales efforts across an organization’s multiple channels. The key entities included with this module are opportunities, opportunity products, competitors, and decision issues.

Siebel Service Master

The Siebel Service Master module is a data model extension that stores an organization’s service information. This extension creates a central storage location for service efforts across an organization’s multiple channels. The key entities included with this module are service requests, solutions, and account and policy details.

Siebel eBusiness Applications with Siebel UCM Applications

Certain product components of Siebel eBusiness Applications are included with Siebel UCM Applications and provide, or add, functionality to the master files. Only the Siebel products required to run Siebel UCM Applications are included with the Siebel UCM Applications base product. The eBusiness Application products outlined below are included with Siebel UCM Applications:

- Siebel Application Object Manager

  Siebel Application Object Manager hosts the Business Objects layer and the Data Objects layer of the Siebel eBusiness Application Architecture. See *Siebel Server Administration Guide* for further information.

- Siebel eBusiness Application Integration (Siebel eAI)

  Siebel eAI provides components for integrating Siebel eBusiness Applications with external applications and technologies. See *Overview: Siebel eBusiness Application Integration Volume I* for further information.

- Siebel Enterprise Integration Manager (Siebel EIM)

  Siebel EIM manages the exchange of data between Siebel database tables and other back-office databases. Siebel EIM is used for bulk data imports, exports, merges, and deletes. See *Siebel Enterprise Integration Manager Administration Guide* for further information.

- Siebel Business Process Designer

  Siebel Business Process Designer is a customizable business application that allows you to design, manage, and enforce your business processes. It allows you to design complex workflow processes and automate the enforcement of business policies and procedures. See *Siebel Business Process Designer Administration Guide* for further details about this product.

Based on the software purchased, one or more of the following Siebel products and functionality may also be included:

- Siebel Replication Manager

  Siebel Replication Manager distributes full schema and data replicas in order to support subordinate Siebel deployments. See *Siebel Remote and Replication Manager Administration Guide* for further information.

- Siebel Assignment Manager

  Siebel Assignment Manager distributes and assigns entities such as opportunities or service requests to individuals, positions, or organizations based on defined assignment rules. See *Siebel Assignment Manager Administration Guide* for further information.

- Siebel Tools

  Siebel Tools is an integrated environment for configuring aspects of Siebel eBusiness Applications. See *Siebel Tools Reference* for further information.

- Siebel Analytics

  Siebel Analytics performs an analysis of customer, marketing, product, sales, and service activities based on premapped variables. See *Siebel Analytics Server Administration Guide* for further information.

- Siebel Data Quality

  Siebel Data Quality assists enterprises by standardizing their contact, account, and prospect data by data matching and data cleansing. See *Siebel Data Quality Administration Guide* for further information.

- Siebel Audit Trail Functionality

  Siebel Audit Trail functionality creates a history of the changes that have been made to various types of information in Siebel eBusiness Applications. See *Applications Administration Guide* for further information.

- Siebel D&B Integration

  D&B's information and technology solutions help businesses find profitable customers, reduce credit risk, manage receivables, and manage vendors. D&B's database of commercial information consists of over 75 million records world wide. Siebel D&B integration allows Siebel eBusiness Applications to access and use D&B data and reports. See *Siebel Reports Administration Guide* and *Applications Administration Guide* for further information.

Siebel UCM Applications Scenarios 2

This chapter examines a fictional financial institution, its business requirements, and the Siebel Universal Customer Master Applications (Siebel UCM Applications) product solution based on two different scenarios: one with no Siebel eBusiness Application deployments and one with several Siebel eBusiness Application deployments. For background information about Siebel UCM Applications, see Chapter 1, “About Siebel UCM Applications.”

Siebel Universal Customer Master Scenario 1

The following business scenario is designed to provide guidance for a potential Siebel UCM Applications deployment. The business institution and its existing implementation are described, followed by the benefits of the Siebel UCM Applications and Siebel Universal Application Network solution. A review of the integration process with back-office systems is also included.

Business Client

A large, successful financial institution has evolved into a national enterprise with offices and business units in several locations across the country. Its core banking systems include client deposits, loans, and mortgages. It also offers credit card services and an investment banking service. Through mergers and acquisitions, the financial institution is also involved with the insurance industry, providing life, home, and car policies for its clients.

Existing Implementation

With this large number of business ventures and offices, the financial institution found itself using a myriad of applications for individual business units. Even in the same business units, different departments (HR and Accounting, for example) functioned on different back-office systems, which included multiple front office applications and multiple middle-tier systems, and various applications that stored subsets of customer information. Client, partner, and product information is spread across multiple IT systems. A large percentage of the IT budget for this institution is spent on maintenance and integration of these applications. However, removing or expanding the functionality of these legacy systems is difficult and costly.

Business Solution

The solution is to continue to use the Siebel UCM Applications as a key component in the synchronization of customer profile information across customer-related applications, including back-office. Siebel UCM Applications provide this solution with the following benefits:

- Siebel UCM Applications can be integrated with Siebel Universal Application Network Customer Lifecycle Management processes to reduce integration costs.
- Siebel UCM Applications provide validated solutions with industry leading data quality vendors.
- Siebel UCM Applications make customer data available to systems across the enterprise.
- Siebel UCM Applications allow real-time access to data across multiple applications and platforms.
- Siebel UCM Applications reconcile and synchronize data to provide a unified view of replicated data.
- Siebel UCM Applications provide high performance, availability, and durability.
- Siebel UCM Applications are extensible to allow for new business processes and units.
- Siebel UCM Applications make sure the quality and uniformity of data are reflected across the entire organization.
- Siebel UCM Applications support the addition of newly acquired systems with the integration of customer data.

Siebel UCM Applications Deployment

In this scenario, Siebel UCM Applications operate as a stand-alone application acting without a Customer Relationship Management (CRM) deployment in an application network. Siebel UCM Applications reside outside any operational application and serve to store, synchronize, and reconcile customer data across the financial organization's enterprise. See Figure 1 for a model of this deployment.

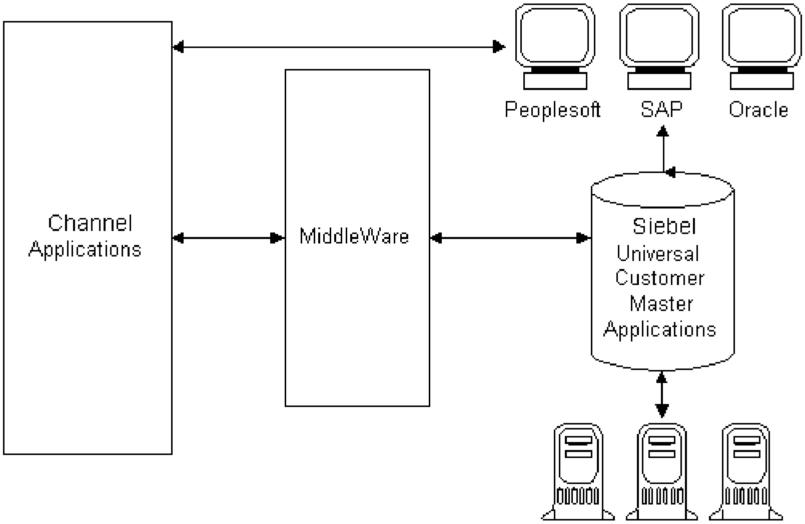


Figure 1. Siebel UCM Applications Stand-Alone Deployment Scenario

Siebel UCM Applications Integration

The Siebel UCM Applications deployment uses several prebuilt connectors along with the full Siebel eAI module functionality to integrate and consolidate customer information across back-office systems. See "Siebel Connector and Integration of Siebel UCM Applications" on page 43 for details about this process. This scenario can also be deployed within a Universal Application Network (UAN) framework.

Siebel Universal Customer Master Scenario 2

Using the same fictional financial institution with the same business conditions discussed in the previous business scenario, the second scenario examines the case in which existing Siebel eBusiness Applications systems are among the various systems that make up the financial institution's enterprise. The Siebel UCM Applications deployment changes in this scenario, but the business solution remains the same.

Siebel UCM Applications Deployment

In this scenario, if the Siebel UCM Applications deployment works with one Siebel eBusiness Application, as well as with other back-office applications, the Siebel eBusiness Application relies on Siebel UCM Applications as the Siebel database. Siebel UCM Applications use middleware infrastructure to consolidate other customer data from back-office systems. See Figure 2 for a model of this deployment.

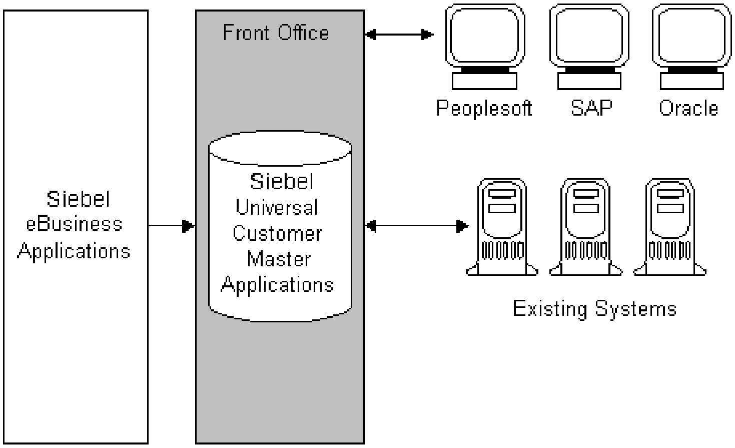

Siebel eBusiness Applications
Front Office
Siebel Universal Customer Master Applications
Peoplesoft
SAP
Oracle
Existing Systems

Figure 2. Siebel UCM Applications Deployed with a Siebel eBusiness Application

Alternatively, if Siebel UCM Applications work in a deployment including multiple instances of Siebel eBusiness Applications, as well as other back-office systems, Siebel UCM Applications act as the central database of enterprise information for all systems. See Figure 3 for a model of this deployment.

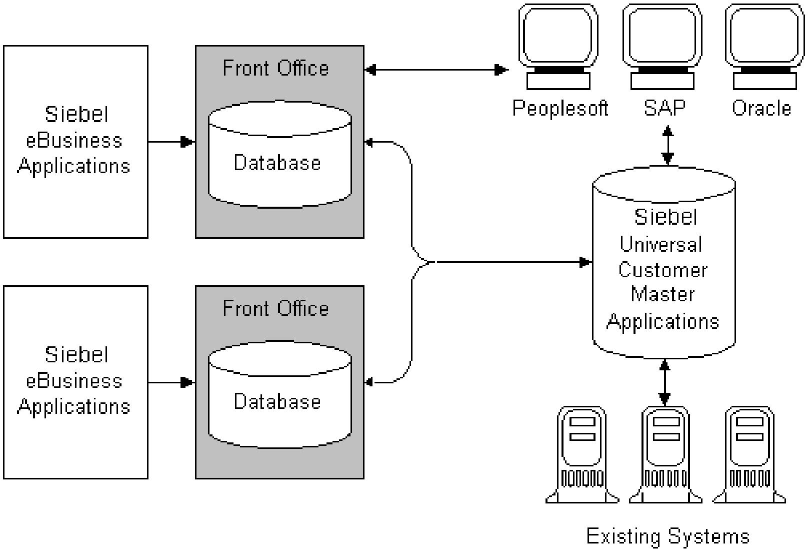

Siebel eBusiness Applications
Front Office
Database
Peoplesoft
SAP
Oracle
Siebel Universal Customer Master Applications
Siebel eBusiness Applications
Front Office
Database
Existing Systems

Figure 3. Siebel UCM Applications Deployed with a Multi-Siebel eBusiness Application Environment

Siebel UCM Applications Integration

As in the previous scenario, the Siebel UCM Applications deployment uses the full Siebel eAI module functionality to integrate and consolidate customer information across back-office systems. Additionally, in a deployment with one or more Siebel eBusiness Applications, the Siebel Connector for Siebel eBusiness Applications can be used to share data with Siebel UCM Applications. See "Siebel Connector and Integration of Siebel UCM Applications" on page 43 for further information.

Architecture and Framework 3

This chapter provides a background for the framework, architecture, integration, and interactions of Siebel Universal Customer Master Applications (Siebel UCM Applications). It also briefly describes the application's functionality and administration.

NOTE: This chapter uses former naming conventions for the Siebel UCM Applications product when referencing hard-coded product business services. References to FINS CIF (Financial Services Customer Information File) are valid for the Siebel UCM Applications for Industry Applications product.

Framework and Interactions of Siebel UCM Applications

Siebel UCM Applications form the master application and database of an organization's data. It must interact with back-office systems and Siebel deployments to provide different organizational business units with consistent and timely data. A variety of functionality is available between Siebel UCM Applications and Siebel eBusiness Applications or back-office applications using technical integration and the regular links to Siebel eBusiness Applications. See Figure 4 for this relationship.

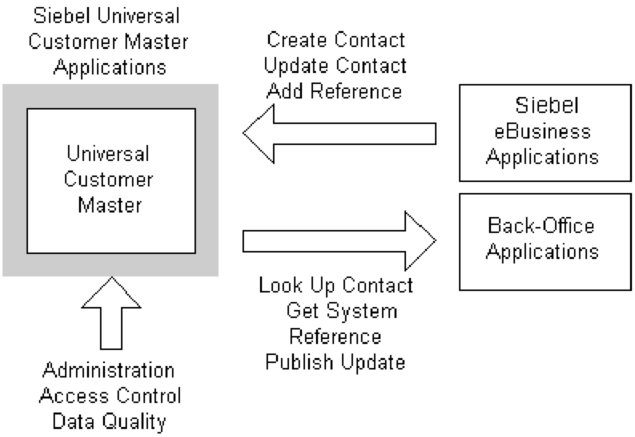


Figure 4. Siebel UCM Applications Framework

- For information about the functionality included with Siebel UCM Applications, see "Functions and Services of Siebel UCM Applications" on page 27.
- For information about the administrative functionality of Siebel UCM Applications, see "Administration of Siebel UCM Applications" on page 28.
- For information about the technical integration aspects used with Siebel UCM Applications, see "Siebel Connector and Integration of Siebel UCM Applications" on page 43.

Functions and Services of Siebel UCM Applications

The database tables and Siebel Connector for Universal Customer Master Applications support insert, delete, update, and query functionality.

- Insert—allows the insertion of new data records into Siebel UCM Applications.
- Delete—allows the deletion of data records from Siebel UCM Applications.
- Update—allows the modification of data records in Siebel UCM Applications.
- Query—allows queries for subsets of data records from Siebel UCM Applications.

If these functions are available in Siebel eBusiness Applications and back-office systems, they can be used to make modifications to an organization's data, which is then stored and reconciled in Siebel UCM Applications.

Siebel UCM Applications also provide the following services through configuration:

- Publish—publish data records from Siebel UCM Applications.
- Request and Reply—request data from Siebel UCM Applications and return data.

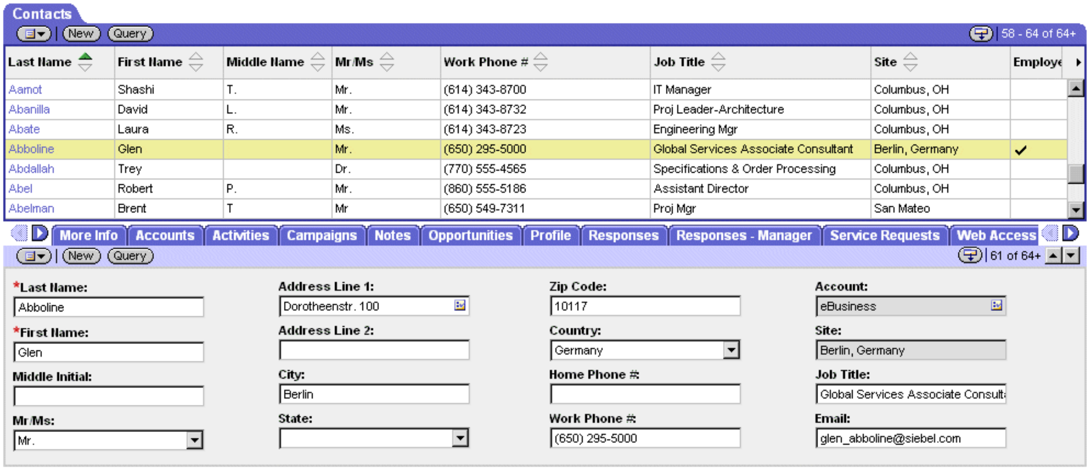
Services can be configured to operate in real time or batch mode.

Siebel UCM Applications use a variety of integration techniques to allow the operation of functions and services between other applications and themselves. See "Architecture of Siebel UCM Applications" on page 30 for further information.

CAUTION: A conflict resolution feature is not currently available in Siebel UCM Applications.

Administration of Siebel UCM Applications

Siebel UCM Applications are administered through Siebel eBusiness Application user interface screens. However, only those screens necessary for viewing and administering the information contained within Siebel UCM Applications are included. See Chapter 4, “Siebel UCM Applications Administration” for further information on administration. Only administrative users have access to these screens. An example of the Contacts screen appears in Figure 5.



Last Name	First Name	Middle Name	Mr/Ms	Work Phone #	Job Title	Site	Employe
Aamot	Shashi	T.	Mr.	(614) 343-8700	IT Manager	Columbus, OH	
Abanilla	David	L.	Mr.	(614) 343-8732	Proj Leader-Architecture	Columbus, OH	
Abate	Laura	R.	Ms.	(614) 343-8723	Engineering Mgr	Columbus, OH	
Abboline	Glen		Mr.	(650) 295-5000	Global Services Associate Consultant	Berlin, Germany	✓
Abdallah	Trey		Dr.	(770) 555-4565	Specifications & Order Processing	Columbus, OH	
Abel	Robert	P.	Mr.	(860) 555-5186	Assistant Director	Columbus, OH	
Abelman	Brent	T	Mr	(650) 549-7311	Proj Mgr	San Mateo	

*Last Name: Abboline
*First Name: Glen
Middle Initial:
Mr/Ms: Mr.
Address Line 1: Dorotheenstr. 100
Address Line 2:
City: Berlin
State:
Zip Code: 10117
Country: Germany
Home Phone #:
Work Phone #: (650) 295-5000
Account: eBusiness
Site: Berlin, Germany
Job Title: Global Services Associate Consult
Email: glen_abboline@siebel.com

Figure 5. Siebel eBusiness Applications Contacts Screen

The administration screens are used to look up small subsets of data.

Access Control is enforced at the user interface level for Siebel UCM Applications administration screens. Siebel Universal Customer Master Application messages have full visibility of the data and run through the privilege check when accessing the Universal Customer Master Application. For further information and procedures, see *Siebel Tools Reference* and *Security Guide for Siebel eBusiness Applications*.

Integration Services for Siebel UCM Applications

The Siebel Connector for Universal Customer Master Applications provides integration between Siebel UCM Applications and other XML-based applications that support Siebel Universal Customer Master Application business process specifications. The Siebel Connector for Universal Customer Master Applications is a configurable set of components, which you can use to exchange data between Siebel eBusiness Applications and external applications and databases.

The Siebel Connector for Universal Customer Master Applications is responsible for receiving, parsing, and executing the business processes specified in the XML message that conforms to the specification detailed in the Siebel Universal Customer Master Application business process specification document. This solution allows you to manage information between Siebel Universal Customer Master Application applications and other applications. The Siebel Connector for Universal Customer Master Applications extends Siebel applications to integrate with back-office data and business processes by serving as a master application for all other applications.

The Siebel Connector for Universal Customer Master Applications supports both synchronous and asynchronous transactions across application boundaries. The resulting data consistency and the ability to share information allows for efficient coordination among enterprise operations. See Chapter 2, “Siebel UCM Applications Scenarios” for scenarios using the Siebel UCM Applications.

In addition, the Siebel Connector for Universal Customer Master Applications also offers the following capabilities:

- Registration and administration of systems accessing the Universal Customer Master Application.
- Enforcement of execution privileges for systems accessing Universal Customer Master Application.
- Broadcasting of customer information changes to enterprise systems.
- Retrieval of External IDs from the Key Map.
- Generate Universal Unique IDs (UUID)

These capabilities are not possible without a predefined message structure to relay required information in exchanged XML messages known as the CRMML Standard.

Architecture of Siebel UCM Applications

Siebel UCM Applications are based on the Siebel eBusiness Application n-tier object architecture in which the user interface, business logic, and data are separated and layered, see Figure 6. Each tier contains a set of objects and components, which allows a high degree of reuse. The consolidated object layer makes for uniform communication across channels and interfaces. This architecture also allows flexible deployment scenarios and integration with new business processes and systems.

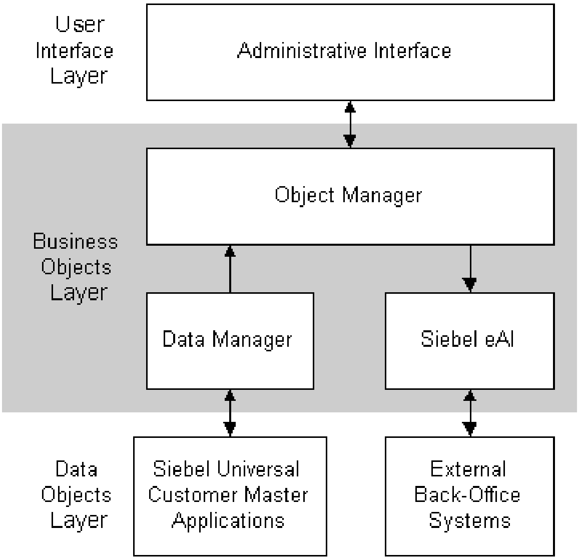


Figure 6. Siebel UCM Applications n-Tier Architecture

The Siebel UCM Applications platform is designed for extreme fault tolerance with zero downtime. The database platform includes:

- High-availability upgrades and online upgrades of the production environment.
- Enhanced cluster support that allows automatic and transparent fail-over between servers and eliminates single points of failure.

- Database auto-reconnect, which allows servers to continue operation after a temporary outage without any intervention.
- Server component reconfiguration without stopping any current tasks or client sessions.
- Supports multithreaded and interactive components, except background mode components.

User Interface Layer

The user interface layer for Siebel UCM Applications is interactive and customizable. It consists of two parts: the physical user interface layer, which includes templates and tags that render the user interface, and the logical user interface objects layer, which presents the data in the user interface. The physical rendering includes applets, charts, and reports. There are multiple presentation formats, including HTML, XML, and WML. The user interface layer contains no business logic. For further information about the user interface layer, see the Siebel Architecture Basic Concepts chapter in *Siebel Tools Reference*. The user interface layer is used for the administration of the Siebel eBusiness Information suite; see "Administration of Siebel UCM Applications" on page 28 for further information.

Business Objects Layer

The business logic layer for Siebel UCM Applications contains object abstractions of entities and represents internal and external data. The logic layer also allows for complex entity relationships. For further information about the business objects layer, see the Siebel Architecture Basic Concepts chapter in *Siebel Tools Reference*.

The business object manager consists of the Object Manager, Data Manager, and Siebel UCM Applications Integration.

Object Manager and Data Manager

The object manager and data manager layer use:

- Business Objects
- Business Components
- Virtual Business Components

These business objects are customizable. For further information about the business objects and data objects layer, see *Siebel Tools Reference*.

Siebel Industry eAI Architecture

Siebel Industry eAI architecture is built to work with the Siebel eBusiness Application Integration (eAI) architecture and to support XML-based messaging communication infrastructure. Different applications require integration using messaging mechanisms. Connectors must be built to support various industry standards.

Siebel eBusiness Applications allow you to build and deploy multiple connectors. For example, the Siebel Financial Services Application has built three connectors—Siebel Connector for Universal Customer Master Applications, Siebel Connector for ACORD XML, and Siebel Connector for IFX XML—based on the Siebel Industry eAI framework.

Figure 7 illustrates the high-level architecture of the Siebel Industry eAI and the standard connectors.

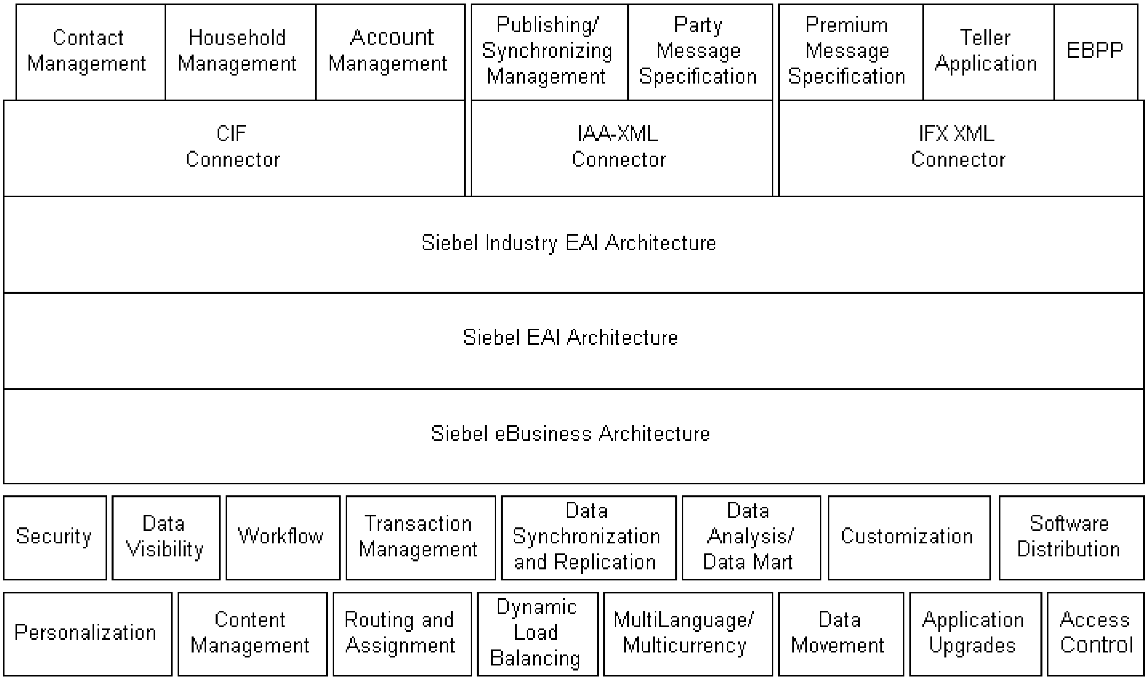


Figure 7. High-Level Architecture of Siebel Industry eAI and Standard Connectors

Siebel Connector for Universal Customer Master Applications is based on Siebel XML standard in addition to a predefined Universal Customer Master Application envelope, headers for identifying the sending system, and associated privileges information. It is designed to address the real-time requirement by defining business processes that include both a request and a response message. The Siebel Connector for Universal Customer Master Applications provides functions such as:

- Handling message header
- Handling heterogeneous objects in the body section of an XML message
- Security checking for system registration

- Privilege checking for customer business process management
- Publishing customer information changes in both real and batch time processing

The Siebel Connector for Universal Customer Master Applications includes the FINS CIF Dispatcher, FINS CIF Converter, FINS CIF Publish/Subscribe Service, FINS CIF Security Manager, and FINS CIF Transaction Manager.

Business Process Flow

Each standard integration or custom integration is based on the creation of business process flows. A business process flow controls the entire business transaction instance. Figure 8 illustrates inbound (Receive-Send) business process flow. Some of the business flows might constitute messages published by Siebel UCM Applications such as <InsertContactProduct>, <DeleteAccountAddress>, <UpdateContactProduct>, <UpdateHousehold>, <DeleteAccountAddressResponse>, <LookUpAccount>, <LookUpPersonalAddressByState,> and so on. These messages are included in Siebel UCM Applications for Industry Applications.

The processing flow for each of these business process flows is largely contained within a Siebel workflow process. The workflow process is instantiated by the Business Integration Manager after receiving the inbound request from enterprise applications.

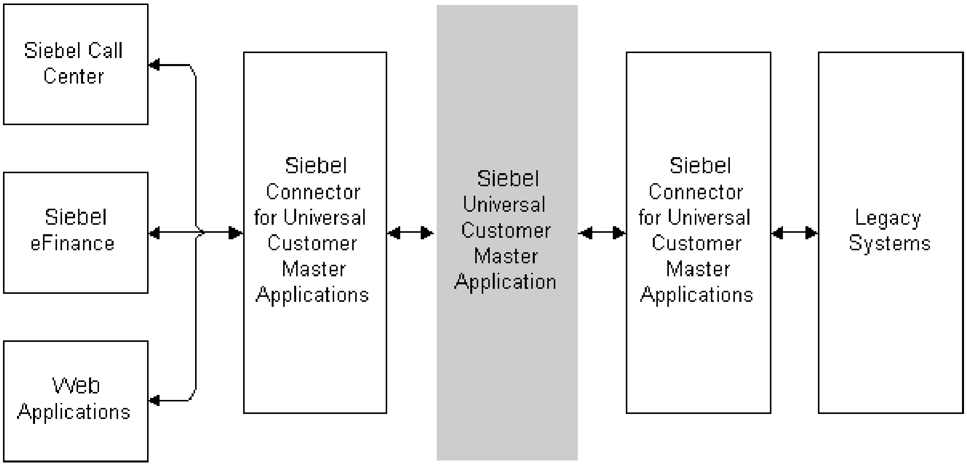

Siebel Call Center
Siebel eFinance
Web Applications
Siebel Connector for Universal Customer Master Applications
Siebel Universal Customer Master Application
Siebel Connector for Universal Customer Master Applications
Legacy Systems

Figure 8. Business Process Flow

Inbound Data Flow

As illustrated in Figure 8 on page 35, an inbound business process flow starts with a Receiver Server Component, such as the MQSeries or HTTP Receiver. The Receiver runs in the background, continuously waiting for messages to arrive from external applications. After receiving a CRMML message, the receiver then invokes the workflow process configured to handle and process the data.

The workflow dictates the business logic behind the Siebel Connector for Universal Customer Master Applications as follows:

1. The raw XML text string passes through XML Hierarchy Converter and is converted into an XML integration object instance.

2 The FINS CIF Dispatcher traverses the XML instance and identifies the messages received according to the rule sets of the CIF Dispatcher Map. It then identifies the envelope, header, and body sections among the hierarchy nodes and sends it to the FINS CIF Security Manager.

NOTE: The Dispatcher Map is shipped as a part of Siebel UCM Applications. For details, see "CIF Dispatcher Map Integration Objects" on page 75.

3 The FINS CIF Security Manager takes the XML instance, verifies the System for registration with Siebel UCM Applications, checks the operation identified in each body section for privileges, and attaches any fail privilege bodies to the FailSecurityHierarchy before sending it to the FINS CIF Converter.

4 The FINS CIF Converter takes the XML instance, and processes individual sections of the instance while converting each sub-tree into external integration object instances before sending it to the FINS CIF Transaction Manager.

5 The FINS CIF Transaction Manager performs operations specified in the instance by invoking the services configured in its user properties. For Insertion messages, a Unique ID is generated and inserted into the <Party UId> tag prior to the operation, and a cross-reference record is created after a successful operation of the instance.

6 If FINS CIF Publish/Subscribe Service is configured for real-time publishing, then messages are sent to systems registered for the particular business objects executed.

Workflow Integration

Siebel Business Process Designer is the center of the business data flow. Workflow processes control the flow and transformation of data into and out of the Siebel eBusiness Applications. You create the workflow processes using a graphical user interface provided within the Siebel eBusiness Applications, called the Siebel Business Process Designer.

For details on workflow and the Siebel Business Process Designer, see *Siebel Business Process Designer Administration Guide.*

Integration Objects

Integration objects are data containers used within the workflow environment. They represent the data structure of a Siebel Business Object or an external application's data object. You create integration objects with the Integration Object Builder provided in Siebel Tools. The Integration Object Builder creates Siebel Integration Objects from Siebel Business Objects, which are then used by components within the Siebel UCM Applications Integration. Supporting integration objects for Siebel UCM Applications business processes are shipped as part of the Siebel UCM Applications.

For more information on Integration Objects, see *Overview: Siebel eBusiness Application Integration Volume I*.

Business Services

Business services execute predefined or custom actions in a workflow process. Examples of business services include the FINS CIF Transaction Manager, the Siebel EAI Adapter, the FINS CIF Converter, and so on. These business services act on property sets passed to them. They perform business logic operations such as interfacing with the database, interfacing to external systems, or transforming one integration object into another. Many business services are provided, but you can create your own. Although you can use business services to perform many different functions, they all have a standard interface. Business services have object-like qualities, such as methods, method arguments, and user properties. These elements define how a business service can be used. Business services are defined in Siebel Tools. This guide describes those business services used to interface to external systems within an organization's application network.

For more information on business services in general, *Integration Platform Technologies: Siebel eBusiness Application Integration Volume II*.

CRMML Standard

Customer Relationship Management Markup Language (CRMML) is a Siebel messaging system targeted for CRM data exchange and remote business process invocation. CRMML allows different applications with different data models to share information, or replicate data through document exchange and cross reference objects. It also allows multiple applications to share business logic and business processes through remote procedure calls.

CRMML is an extensible XML vocabulary. CRMML specification defines a set of syntax and rules to describe how to interpret messages and how to extend the messages beyond the current offer. CRMML messages contain a simple envelope, a header section, and a body section. The details are described in the following sections.

CRMML Syntax and Rules. CRMML messages contain an envelope, a header section, and a body section. The message specification focuses on the hierarchy and definitions of header elements. It allows users to extend the body section with user defined elements as long as the object descriptors in the header section describe the usages of the elements in a body section. The following example illustrates the layers of a CRMML message:

```
<?xml version="1.0" encoding="UTF-8>
<CRMML>
<! CRMML Header section starts>
      <CRMHeader>
            <FromSystem>
                  <SystemInfo type="ID">
                  <SystemId>www.mycompany.com</SystemId>
                  <SystemRole>Insurance Application</SystemRole>
                  <SystemName>Client UI System</SystemName>
                  </SystemInfo>
            </FromSystem>
            <ToSystem>
                  <SystemInfo type="ID">
                  <SystemId>123-456-789</SystemId>
                  <SystemRole>Service Provider</SystemRole>
                  <SystemName>Siebel CIF System</SystemName>
                  </SystemInfo>
            </ToSystem>
            <MessageDescriptor>
                  <MessageId>11SC3I2002053116:43:45</MessageId>
                  <TransactionScope>any</TransactionScope>
                  <Timestamp>01/01/2001</Timestamp>
                  <TimeToLive>365</TimeToLive>
            </MessageDescriptor>
      </CRMHeader>
<!CRMML Header section ends>

      <CIFPartyPackage>
         <InsertContactProfile>
               <Contact>
                     <Id>ext2223</Id>
                     <EmailAddress>cifemail</EmailAddress>
                     <FaxPhone>5555555555</FaxPhone>
                     <FirstName>0099INFN</FirstName>
                     <HomePhone>1111111111</HomePhone>
                     <LastName>0099INLN</LastName>
               </Contact>
         </InsertContactProfile>
      </CIFPartyPackage>
</CRMML>
```

Envelope and Header Section. The CRMML message has one envelope element <CRMML>. <CRMML> is the root element for a message. It contains one optional header element <CRMHeader> and one or more body elements. The envelope has no attributes.

The header includes elements for sender and receiver. The following is an example for communication parties:

```
<! CRMML Header section starts>

<CRMHeader>

   <FromSystem>
      <SystemInfo type="ID">
            <SystemId>www.mycompany.com</SystemId>
            <SystemRole>Insurance Application</SystemRole>
         <SystemName>Client UI System</SystemName>
      </SystemInfo>
   </FromSystem>
   <ToSystem>
      <SystemInfo type="ID">
         <SystemId>123-456-789</SystemId>
         <SystemRole>Service Provider</SystemRole>
         <SystemName>Siebel CIF System</SystemName>
      </SystemInfo>
   </ToSystem>
   …
<CRMHeader>
```

<FromSystem> is the sender and <ToSystem> is the receiver in this example. Both elements contain <SystemInfo> aggregate, which holds system ID, system role, and the name of the system. In the example, Sender's name is Client UI System and receiver is Siebel UCM Applications System. Sender plays an insurance application role and receiver plays service provider's role.

The header also contains a message descriptor that depicts the body of the message and objects inside the body sections. An example of the message descriptor follows:

```
<CRMML>
<! CRMML Header section starts>
      <CRMHeader>
            <MessageDescriptor>
                  <MessageId>11SC3I2002053116:43:45</MessageId>
                  <TransactionScope>any</TransactionScope>
                  <Timestamp>Sunday</Timestamp>
                  <TimeToLive>OneYear</TimeToLive>
            </MessageDescriptor>
      </CRMHeader>
```

The <MessageId> element can be unique to identify a single message, a pair of request or response messages, or a group of communication messages. Transaction scope can be any or all. <Timestamp> is for the creation time of a message and <TimeToLive> is the expiration time of a message. Receiver should disregard a message if a message is expired.

The <MessageDescriptor> can have zero or more <ObjectDescriptor> child elements. The following object descriptor example illustrates a metadata for remote procedure call in a message header:

```
<CRMHeader>
      <MessageDescriptor>
      …
      <ObjectDescriptor>
            <ObjectId type=”HREF” sytem=”this”>p1</ObjectId>
            <ObjectUsage>SoftwarePackage</ObjectUsage>
      </ObjectDescriptor>
      <ObjectDescriptor>
            <ObjectId type=”HREF” sytem=”this”>m1</ObjectId>
            <ObjectUsage>Method</ObjectUsage>
      </ObjectDescriptor>
      <ObjectDescriptor>
            <ObjectId type=”HREF” sytem=”this”>a1</ObjectId>
            <ObjectUsage>Argument</ObjectUsage>
      </ObjectDescriptor>
      </MessageDescriptor>
</CRMHeader>
<CIFPartyPackage ID=”p1”>
      <InsertContactProfile ID=”m1”>
            <Contact ID=”a1”>
```

```
                <Id>ext2223</Id>
```

The first <ObjectDescriptor> refers to <CIFPartyPackage> element in the message body by HREF. It indicates that <CIFPartyPackage> is a software package. In this example, it is the Siebel Universal Customer Master Application. The second descriptor indicates that <InsertContactProfile> is a method in the package that the message intends to invoke. The third descriptor shows that <Contact> aggregate is the input argument for <InsertContactProfile>. For CIF implementation, the default usages of the first three elements are package, method, and argument; therefore, no objective descriptor is required for Universal Customer Master Application specific messages.

Body Section. Body section contains user defined message elements. The following is an example body section and is one of the messages supported by the Siebel UCM Applications.

```
<CRMML>
      <CRMHeader>
      …
      </CRMHeader>

      <CIFPartyPackage ID=”p1”>
            <InsertContactProfile ID=”m1”>
                  <Contact ID=”a1”>
                        <Id>ext2223</Id>
                        …
```

The example indicates that an external application requests to insert a contact profile to the Siebel UCM Applications database. The contact detail information is contained within the <Contact> aggregate.

Data Objects Layer

The Data Objects layer for Siebel UCM Applications separates other layers from the database and is used only for data storage. It has native SQL and fully dynamic SQL generation. The data layer allows runtime switching between databases, for example, from local to server or testing to production. It auto-adapts to object manager and user interface requirements, and requires zero downtime for major release schema upgrades.

Siebel UCM Applications support leading databases (DB2 UDB for Windows and UNIX, DB2 UDB for OS/390 and z/OS; Oracle; Microsoft SQL Server) on various platforms. The data layer can scale vertically (with size of server) and horizontally with database parallelization. For further information about the Data Objects layer, see the Siebel architecture basic concepts chapter in *Siebel Tools Reference*.

Siebel Connector and Integration of Siebel UCM Applications

Siebel Connector for Information File Applications, known as Siebel Connector for Universal Customer Master Applications, describes a set of configurable components for integration among enterprise applications and Siebel UCM Applications.

The Siebel Connector for Universal Customer Master Applications comprises five prebuilt business services:

- FINS CIF Transaction Manager
- FINS CIF Converter
- FINS CIF Dispatcher
- FINS CIF Security Manager
- FINS CIF Publish/Subscribe Service

FINS CIF Transaction Manager

The FINS CIF Transaction Manager executes operations specified in Universal Customer Master Application message instances as Siebel database transactions. The FINS CIF Transaction Manager also:

- Evaluates heterogeneous commands and executes the transactions.
- Invokes business services, including Siebel eAI, that are configured in its user properties. These business services can be invoked multiple times.
- Translates Universal Customer Master Application command elements to Siebel Adapter actions and combines return results as a single property set.

Table 1 lists the user properties for FINS CIF Transaction Manager.

Table 1. User Properties of FINS CIF Transaction Manager

Name	Value	Description
DispatcherMapName	CIFDispMap	Transaction manager uses this map to tag the Body information for other components. This value can be set as a runtime input argument as well.
IgnoreSvcMethodArgs	true or false	Allows runtime input arguments.
XXX (Operation)	ServiceName/ MethodName/Argument	Type of operation to use. For details, see "FINS CIF Transaction Manager" on page 43.

The FINS CIF Transaction Manager uses the pre-built methods described in Table 2 to process inbound messages.

Table 2. Pre-built Methods for FINS CIF Transaction Manager

Name	Value	Description
Execute	Execute Transaction	Used for inbound or outbound purposes, as long as the integration object instance is provided. Note that you should use the Execute Outbound method when Row ID is the only available input.

FINS CIF Transaction Manager business service uses combinations of user property, method and method arguments to achieve different tasks. Table 3 lists the method arguments available to use with FINS CIF Transaction Manager business service.

Table 3. Method Arguments for Use with FINS CIF Transaction Manager

Argument	Default Value	Description
OnlyIOI	FALSE	Used for Inbound integration. The inbound message may contain header, body, and envelope portion. When Transaction Manager takes the proper operation against the Siebel application, the integration object instances for response is generated as well. Information from the request message is dropped if this value were set to true. Therefore, the FINS CIF Converter, and others, do not need to deal with the overhead. If this is not set to true, then request information is still carried over.
XMLHierarchy	N/A	Property set in CIF integration object hierarchy.
TurnOffCIF	true or false	Allows runtime input argument to turn off Siebel UCM Applications specific operations. Setting this parameter to true will cause the CIF Transaction Manager to bypass the Unique ID generation for Party UID feature and the cross-referencing feature.

Although these arguments are available for use by FINS CIF Transaction Manager, not all can be used with each method. Table 4 lists the arguments for each method.

Table 4. Execute Method Arguments

Argument	Display Name	Data Type	Type	Optional
Method: Execute				
OnlyIOI	Only produce Integration Object Instance	String	Input	
StatusObject	Status Object	String	Input	Y
TurnOffCIF	Turn Off CIF features	String	Input	Y
XMLHierarchy	XML Property Set	Hierarchy	Input / Output	

FINS CIF Converter

The FINS CIF Converter converts Siebel UCM Applications messages received to a property set for inbound communication. The converter iterates through the requests and responses in the message to construct any error information (if any), and also constructs the envelope and header section of the message. Table 5 lists the user properties for this converter.

Table 5. User Properties for FINS CIF Converter

Name	Value	Description
EscapeNames	true or false	If set to true, the converter replaces illegal XML name characters with escape characters. Otherwise, it ignores illegal XML name characters. The default value is true.

Table 5. User Properties for FINS CIF Converter

Name	Value	Description
HierarchyFormat	XML Hierarchy	Format of the property set input or output. For internal use.
XMLEnvIntObjectName	<integration object name>	Name of an integration object that defines the content and hierarchy for the envelope and header section of CRMML. The default envelope integration object is Siebel UCM Applications Envelope.

This service provides several ready-to-use methods as described in Table 6.

Table 6. Methods for FINS Converter

Methods	Display Name	Description
PropSetToXMLPropSet	PropSetToXMLPropSet	Converts an integration object hierarchy to XML hierarchy.
XMLPropSetToPropSet	XMLPropSetToPropSet	Converts an XML hierarchy to an integration object hierarchy.
GenerateErrorPropSet	GenerateErrorPropSet	Constructs the XML hierarchy to respond to the external systems when undesired operation or configuration errors occurred.

For each method defined, the FINS CIF Converter business service takes input arguments and produces output arguments. These arguments are described in Table 7.

Table 7. Arguments for the FINS CIF Converter

Argument	Default	Description
Escape Names	True	Indicates whether to remove the escape characters or invalid XML characters from the XML hierarchy.
XML Hierarchy	N/A	Property Set in external integration object or XML hierarchy.
Truncate Field Values	True	Truncate field values.
Contains Inline Attachments	True	Indicates the message contains attachment documents.
ErrorMessageText	N/A	The actual error text to be set to the <ErrorDescription> element in CRMML fault section.
ErrorCode	N/A	The actual error code to be set to the <ErrorId> element in CRMML fault section.
ErrXMLHierarchy	N/A	Input hierarchy will contain the request XML hierarchy to be associated with when generating the respond XML hierarchy. This argument can be ignored. Output hierarchy will contain the XML hierarchy with <ErrorInfo> aggregate generated.
CIFSystemContainer	N/A	Property Set storing the sending system information for constructing response messages.
FailSecurityHierarchy	N/A	Property Set storing the operations which fail the security system check.

Each FINS CIF Converter method uses a group of these method arguments to convert the data for the next service in the integration process. Table 8, Table 9, and Table 10 display the arguments for each method.

Table 8. Argument Specifications for the PropSetToXML Method

Name	Default Display Name	Data Type	Type	Optional
XMLHierarchy	< IgnoreEmptyTag >	Hierarchy	Input/Output	No
FailSecurityHierarchy	FailSecurityHierarchy	Hierarchy	Input	Yes
CIFSystemContainer	CIFSystemContainer	Hierarchy	Input	No

Table 9. Argument Specifications for the XMLPropSetToPropSet Method

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/ Output	No
CIFSystemContainer	CIFSystemContainer	Hierarchy	Output	No

Table 10. Argument Specifications for the XMLToPropSet Method

Name	Default Display Name	Data Type	Type	Optional
ErrorMessageText	Error Message Text	String	Input	No
ErrorCode	Error Code	String	Input	No
XMLHierarchy	Err XML Hierarchy	Hierarchy	Input/ Output	No

FINS CIF Dispatcher

The FINS CIF Dispatcher is responsible for receiving and dispatching inbound messages. It receives the inbound message and scans the message for any operations specified in its rule sets dispatcher map, and associates the integration objects for the connector components. It then parses the envelope of the message, converts it to the header property set, and passes the integration objects for the specified action, the envelope layer property set, and the XML message to the connector components for further processing. Table 11 lists the user properties for FINS CIF Dispatcher.

Table 11. User Properties for FINS CIF Dispatcher

Name	Value	Description
DispatcherMapName	< integration object name >	Name of an integration object that details the dispatching rules and syntax for the CRMML standard. This map is usually created along with the other integration objects by the wizard. The default map name is “CIFDispMap”.
XMLEnvIntObjectName	< integration object name >	Name of an integration object that defines the content and hierarchy for the envelope and header section of CRMML. The default envelope integration object is “CIF Envelope”.
XMLFaultObject_O	< path to fault section or element >	This allows dispatcher to identify a fault section with the first token and further confirm it with the value of 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 CIF Dispatcher business service provides ready-to-use methods and method arguments described in Table 12.

Table 12. FINS CIFDispatcher Methods

Method	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.

For each method defined, the FINS CIF Dispatcher business service takes input arguments and produces output arguments. These arguments are described in Table 13 and Table 14.

Table 13. FINS CIF Dispatcher Method Arguments

Argument	Default	Description
XML Hierarchy	N/A	Property Set in XML hierarchy.

Table 14. Method Arguments for DispatchMessage Method

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

FINS CIF Security Manager

The FINS CIF Security Manager acts as the gatekeeper of the Siebel Universal Customer Master Application. It verifies the System ID in the message header against the System Registrations in the Siebel Universal Customer Master Application before the system can perform its desired operations. After a system passes the registration verification process, the FINS CIF Security Manager cycles through the heterogeneous commands in the body. The FINS CIF Security Manager checks the privilege of the individual operations in the body sections against the privileges in the Customer Information for the particular object and operation. For any fail operation instances in the body, it is removed from the XMLHierarchy and attached to the FailSecurityHierarchy. Table 15 lists the user properties for the FINS CIF Security Manager.

Table 15. User Properties for FINS CIF Security Manager

Name	Value	Description
IXMLOperation_ADD	Insert	Insert operation field Name value pair. Checks the Insert field for privilege if the FINS CIF Dispatcher tags the business process as an IXMLOperation_ADD operation.
IXMLOperation_DELETE	Delete	Delete operation field Name value pair. Checks the Delete field for privilege if the FINS CIF Dispatcher tags the business process as an IXMLOperation_DELETE operation.
IXMLOperation_QUERY	Query	Query operation field key value pair. Checks the Query field for privilege if the FINS CIF Dispatcher tags the business process as an IXMLOperation_QUERY operation.
IXMLOperation_UPDATE	Update	Update operation field key value pair. Checks the Update field for privilege if the FINS CIF Dispatcher tags the business process as an IXMLOperation_UPDATE operation.

The FINS CIF Security Manager processes only inbound messages. Table 16 lists and describes the method that process inbound messages.

Table 16. Process Method for FINS CIF Security Manager

Name	Display Name	Description
SecurityCheck	Security Check	Supports only inbound requests to the Siebel Connector for Universal Customer Master Applications. This method checks the System ID in the Envelope against the Siebel Universal Customer Master Application for System Registration. It also checks the individual business process in the body for privileges to perform operations specified.

FINS CIF Security Manager business service uses a combination of user property and method arguments for its methods to achieve different tasks. Table 17 lists the method arguments available to use with FINS CIF Security Manager business service.

Table 17. Method Arguments for Use with FINS CIF Security Manager

Argument	Display Name	Description
SecurityEnforced	N/A	Use to turn on or off the Security checking process. When this value is set to false the FINS CIF Security Manager will bypass the registration verification process and the operation privilege checking process. FINS CIF Security Manager will perform its security checking operations only when the value is set to true.
XMLHierarchy	N/A	Property set in CIF integration object hierarchy.
FailSecurityHierarchy	N/A	Property set in Siebel UCM Applications integration object hierarchy. It includes failed system registration verification body instances or individual business processes that have no privilege to perform the operation specified.

Table 18 lists the detailed arguments for the method Security Check.

Table 18. Arguments for Method Security Check

Argument	Display Name	Data Type	Type	Optional
Method: SecurityCheck				
SecurityEnforced	Security Enforced	String	Input	N
XMLHierarchy	XML Hierarchy	Hierarchy	Input or Output	N
FailSecurityHierarchy	Fail Security Hierarchy	Hierarchy	Input	Y

FINS CIF Publish/Subscribe Service

The FINS CIF Publish/Subscribe Service supports both Real Time and Daily Batch publish types. The Real Time publish module takes the Siebel Connector for Universal Customer Master Applications output message, which could include heterogeneous body segments with different business object types, as an input message. It dynamically constructs a unique output message for each system based on the Business Object type that the system registered in the Publish/Subscribe administration view. The Real Time publish module sends the constructed messages to the registered systems through a registered protocol type.

Similarly, the Daily Batch publish module publishes the synchronization information of registered Business Object types. The module constructs messages according to Siebel UCM Applications message specification to the registered system through registered protocol types at the registered publish time.

Both publish modules then update the Last Publish time to the system in the Publish/Subscribe table.

The FINS CIF Publish/Subscribe Service has no user properties.

The FINS CIF Publish/Subscribe Service uses the pre-built methods described in Table 19 to publish messages.

Table 19. Method for FINS CIF Publish/Subscribe

Name	Display Name	Description
PublishMethod	Publish	Dynamically routes to Real Time or Daily Batch publish style according to the incoming message style.

FINS CIF Publish/Subscribe Service uses a combination of methods and method arguments to achieve different tasks. Table 20 lists the method arguments available to use with FINS CIF Publish/Subscribe Service business service.

Table 20. Method Arguments for Use with FINS CIF Publish/Subscribe

Argument	Display Name	Description
XMLHierarchy	N/A	Property set in CIF integration object hierarchy. It is the input of real-time publish.
Account	N/A	Input of Account integration object name to be used during Daily Batch publishing time. It is the keyword for identifying the integration object associated with CIFAccount messages.
Contact	N/A	Input of Contact integration object name to be used during Daily Batch publishing time. It is the keyword for identifying the integration object associated with CIFContact messages.
Household	N/A	Input of Household integration object name to be used during Daily Batch publishing time. It is the keyword for identifying the integration object associated with CIFHousehold messages

Although these arguments are available to be used by FINS CIF Publish/Subscribe Service, not all can be used with each method. Table 21 and Table 22 list the arguments for each method.

Table 21. Argument Details for Publish Method (Real Time)

Argument	Display Name	Data Type	Type	Optional
XMLHierarchy	XML Hierarchy	Hierarchy	Input	

Table 22. Argument Details for Publish Method (Daily Batch)

Argument	Display Name	Data Type	Type	Optional
Account	Batch Account Integration Object Name	String	Input	
Contact	Batch Contact Integration Object Name	String	Input	
Household	Batch Household Integration Object Name	String	Input	

NOTE: For Daily Batch, configure one of the integration objects to be used in each batch publish workflow.

Transport Adapter

Transport Adapter is a prebuilt business service providing an interface between Siebel eBusiness Applications and external applications. Transports allow Siebel UCM Applications to exchange data with external applications using standard technologies for both synchronous and asynchronous communication protocols. Transports provide connectivity to virtually any communication protocol that can represent data as text or binary messages, including MQSeries from IBM, and HTTP.

For details on transport adapters, see *Transports and Interfaces: Siebel eBusiness Application Integration Volume III*.

Siebel EIM and Integration of Siebel UCM Applications

Siebel EIM is an important component of Siebel UCM Applications. It bulk imports and exports data residing in other back-office systems into Siebel UCM Applications and consolidates and replicates information from these sources. Siebel EIM provides a schema abstraction and has high throughput. Some features of Siebel EIM include:

- Denormalized interface tables
- Declarative mapping defined in Siebel Tools
- Automated SQL generation
- Support for parallelized import

For further information about Siebel EIM, see *Siebel Enterprise Integration Manager Administration Guide*.

Siebel UCM Applications Administration 4

This chapter details the administrative capabilities of Siebel Universal Customer Master Applications (Siebel UCM Applications). Only users with administration privileges have access to these screens and procedures.

NOTE: This chapter uses former naming conventions for the Siebel UCM Applications product when referencing hard-coded product screens and views. References to CIF (Customer Information File) are valid for the Siebel UCM Applications for Industry Applications product.

System Registration

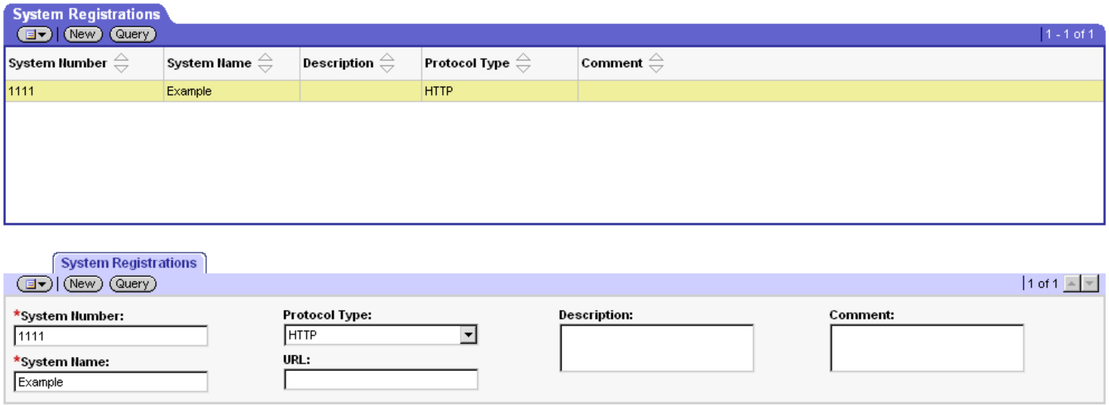
Every application or system that connects to Siebel UCM Applications must register through the System Registration tab of the Customer Information File view. Once registered, the system's privileges and accessibility to the Siebel UCM Applications is administered on other tabs within this view. See "System Privileges" on page 62 and "System Publish and Subscribe" on page 64 for this information.

To create a new system

1. Navigate to the Customer Information File Administration screen.
2. In the System Registration list, click the menu button and then New Record.
3. Enter the appropriate information in the available fields to define the system. Use the following table for information on each of the fields.

Field	Description
System Number	The System Number represents an identifier for an application that accesses Siebel UCM Applications. This number can be an IP address or some other means to identify a system.
System Name	The System Name is the name of the application that accesses Siebel UCM Applications.
Protocol Type	The protocol that an application uses to access Siebel UCM Applications. Currently, this value is HTTP or MQSeries.
Queue Manager Name	Name of the queue manager that receives and sends messages from Siebel UCM Applications. Only applicable when protocol type selected is MQSeries.
Queue Receiver Channel	Name of the queue configured on the Queue Manager to receive requests and send responses. Only applicable when protocol type selected is MQSeries.
URL	URL destination for posting responses to messages. Only applicable when protocol type is HTTP.
Description	Description of the application accessing Siebel UCM Applications.
Comments	Comments regarding the application accessing Siebel UCM Applications.

Figure 9 shows the creation of an example system.



System Number	System Name	Description	Protocol Type	Comment
1111	Example		HTTP	

*System Number: 1111

*System Name: Example

Protocol Type: HTTP

URL:

Description:

Comment:

Figure 9. Creating a new system

To delete a system

1. Navigate to the Customer Information File Administration screen.
2. In the System Registration list, select the system of interest.

   The System Registration form for this system, which appear below the list, provides detailed information on this system.
3. Click the menu button and then Delete Record.

   A confirmation dialog appears.
4. Select OK to delete the system.

System Privileges

After a system is registered (see "System Registration" on page 60 for information on registering a system), administrators can set privileges for this system, which allows grant access to Siebel UCM Applications. For example, a system may allow insert, update, and query privileges on Contact records, but not allow the privilege of deleting the contact. Privileges are granted on the Business Objects layer.

To set a system's privileges

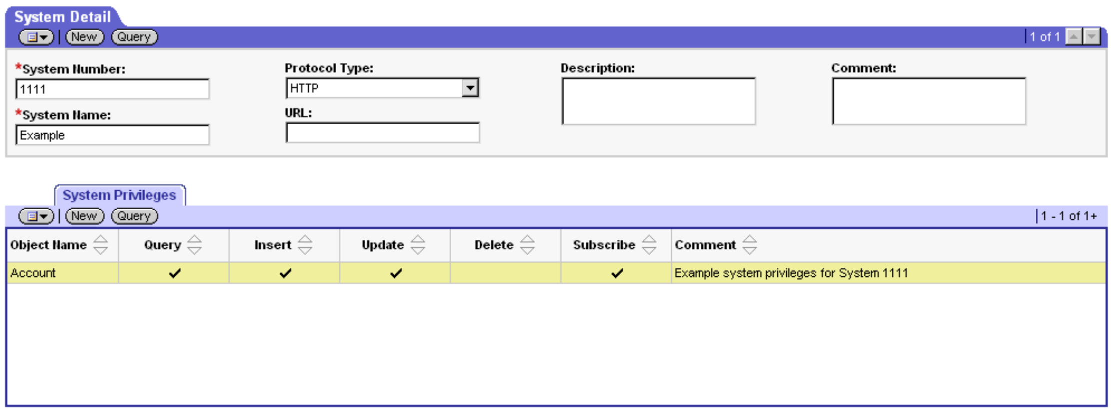
1. Navigate to the Customer Information File Administration screen.
2. In the Systems Registrations list, select the system of interest.
3. From the Show drop-down list, select System Privileges.

   The System Detail and System Privileges tabs appear.
4. In the System Privileges tab, click the menu button and then New Record.
5. Enter the appropriate information in the available fields to define the privileges for each object in the system. Use Table 23 for information on each of the fields.

Table 23. Object Fields

Field	Description
Object Name	An object stored in Siebel UCM Applications like a Contact, Account, Household, and so on.
Query	Query privileges for the object selected in the object name field.
Insert	Insert privileges for the object selected in the object name field.
Update	Update privileges for the object selected in the object name field.
Delete	Delete privileges for the object selected in the object name field.
Subscribe	When selected, the system is set to subscribe to any changes on this particular business object. Subscribe is selected by default. See "System Publish and Subscribe" on page 64 for details.
Comment	Comments on the privileges chosen for the selected object of the selected system.

Figure 10 shows the definition of privileges for the example system.



*System Number: 1111	Protocol Type: HTTP	Description:	Comment:
*System Name: Example	URL:		

Object Name	Query	Insert	Update	Delete	Subscribe	Comment
Account	✓	✓	✓		✓	Example system privileges for System 1111

Figure 10. Defining system privileges

The configuration in Figure 10 indicates that the system registered with the name Example has query, insert, and update privileges for the Account business object. However, it is not allowed to delete any of the Account records in the Siebel Universal Customer Master Application. Furthermore, it is set to subscribe to record changes for the Account business object.

NOTE: A check mark in the Subscribe column (the default setting) does not complete the system publish and subscribing setup steps. Further configuration is required. See "System Publish and Subscribe" on page 64 for further information.

System Publish and Subscribe

The Publish and Subscribe infrastructure provides workflow policies, workflows, and Siebel Universal Customer Master Application messages to publish data records that have been inserted or updated in the Siebel Universal Customer Master Application. The workflow policies track changes to records within the Siebel Universal Customer Master Application and flags them for production; Siebel workflows process the information collected by the workflow policies and guarantee appropriate publication of the changes to subscribed systems; and the messages are predefined and used for the publication of records flagged by the workflow policies.

An external system must be given subscribe privileges to use this feature (See "System Privileges" on page 62 for information on system privileges). The following procedures describe:

- Configuring real-time publish and subscribe for a system
- Configuring batch-time publish and subscribe for a system (see "To configure batch-time publish and subscribe for a system" on page 66).

To configure real-time publish and subscribe for a system

1. Navigate to the Customer Information File Administration screen.
2. In the Systems Registrations list, select the system of interest.
3. From the Show drop-down list, select System Privileges.

   The System Detail and System Publish/Subscribe tabs appear.

4 Enter the appropriate information in the available fields to define the publish and subscribe privileges for each object in the system. Use the following table for information on each of the fields. Make sure to set the Publish Frequency field to real-time.

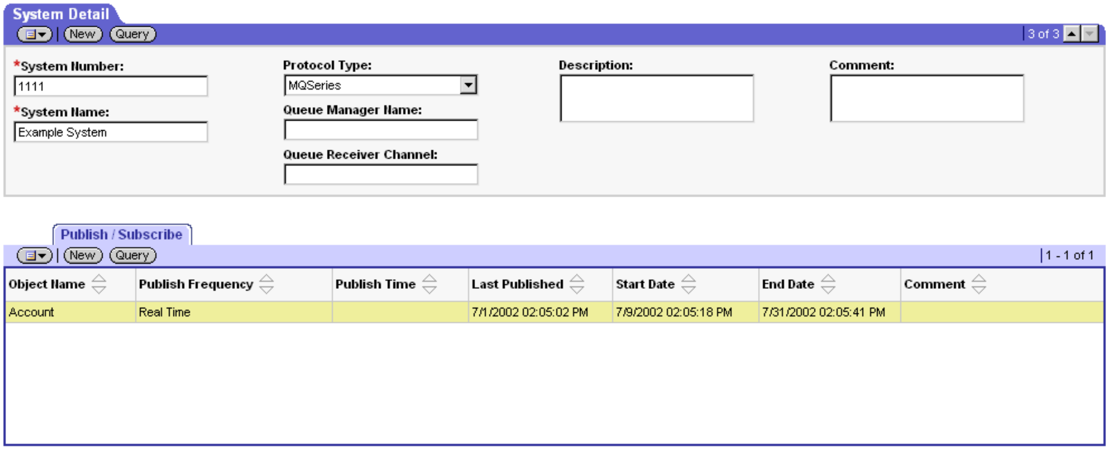
Field	Description
Object Name	An object stored in Siebel UCM Applications like a Contact, Account, Household, and so on.
Publish Frequency	How often the Siebel UCM Applications update the system's information for the selected object. Select the real-time value for real-time publish and subscribe functionality. See "To configure batch-time publish and subscribe for a system" on page 66 for the daily batch procedure.
Last Published	The last time the Siebel UCM Applications publish record changes for the selected object to this selected system.
Start Date	Start date of the publish and subscribe functionality.
End Date	End date of the publish and subscribe functionality.
Comment	Comments on the publish and subscribe definition for the selected object of the selected system.

5 Navigate to the Business Process Administration screen, Workflow Processes view.

6 Query for CIF Server Party Package Workflow or a prepared customized workflow.

7 Make sure that decision point in the workflow process is set to true, which routes incoming messages to the FINS CIF Publish/Subscribe Service. For detailed information on the Siebel Business Process Designer, see *Siebel Business Process Designer Administration Guide.*

Figure 11 displays the definition of real-time publish and subscribe capabilities for the example system.



System Detail

*System Number: 1111

*System Name: Example System

Protocol Type: MQSeries

Queue Manager Name:

Queue Receiver Channel:

Description:

Comment:

Publish / Subscribe

Object Name	Publish Frequency	Publish Time	Last Published	Start Date	End Date	Comment
Account	Real Time		7/1/2002 02:05:02 PM	7/9/2002 02:05:18 PM	7/31/2002 02:05:41 PM	

Figure 11. Defining system real-time publish and subscribe capabilities

To configure batch-time publish and subscribe for a system

1. Navigate to the Customer Information File Administration screen.
2. In the Systems Registrations list, select the system of interest.
3. From the Show drop-down list, select System Privileges.

   The System Detail and System Publish/Subscribe tabs appear.

4 Enter the appropriate information in the available fields to define the publish and subscribe privileges for each object in the system. Use the following table for information on each of the fields. Make sure to set the Publish Frequency field to Daily Batch.

Field	Description
Object Name	An object stored in Siebel UCM Applications like a Contact, Account, Household, and so on.
Publish Frequency	How often the Siebel UCM Applications update the system's information for the selected object. Select the daily batch value for batch-time publish and subscribe functionality. See "To configure real-time publish and subscribe for a system" on page 64 for the real-time procedure.
Publish Time	Selects at which time the update occurs. Only applicable when Publish Frequency is Daily Batch.
Last Published	The last time the Siebel UCM Applications publish record changes for the selected object to this selected system.
Start Date	Start date of the publish and subscribe functionality.
End Date	End date of the publish and subscribe functionality.
Comment	Comments on the publish and subscribe definition for the selected object of the selected system.

5 Navigate to the Business Process Administration screen, Workflow Processes view.

6 Click the menu button and then New Record. Create a new workflow process with any name, such as CIF Account Dailybatch Publish Workflow, see Figure 12. For details on creating workflow processes and the Siebel Business Process Designer, see *Siebel Business Process Designer Administration Guide.*

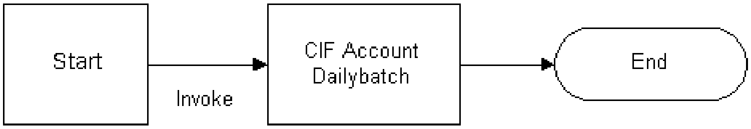


Figure 12. CIF Account Dailybatch Publish Workflow

7 Configure the CIFAccountDailybatch module as follows:

Step	Start
Step	CIF Account Dailybatch
Business Service	FINS CIF Publish/Subscribe Service
Method	Publish
Input Argument	Batch Account Integration Object Name
Type	Literal
Value	CIFContacts (as an example)
Step	End

The integration object name is configurable. You can enter your preconfigured integration object name that corresponds to CIF Account messages that contain the data hierarchy and field in which you are interested. Similarly, if this is a batch workflow for contact or household, enter the integration object name corresponding to CIF Contact messages or CIF Household messages. Then set the input argument to Batch Contact Integration Object Name or Batch Household Integration Object Name respectively.

NOTE: Only one publish message is supported in each workflow for performance. Message size considerations are important as the output of Daily Batch can be large.

8 Configure a repeated component server task to trigger the CIF Batch Workflow on a batch-time basis using the following procedure. See *Siebel Server Administration Guide* for detailed information on this process.

a Navigate to the Server Components Request screen, My Repeating Component Requests view.

b From the Repeating Components Request list, click the menu button and then New Record.

c In the My Repeating Component Request form, enter the following information in the available fields:

Field	Value
Component/Job	Workflow Process Batch Manager
Server	Siebel
Repetitions	Based on your configuration
Repeat Intervals	Based on your configuration
Repeat Unit	Based on your configuration
Repeat From	Actual Start

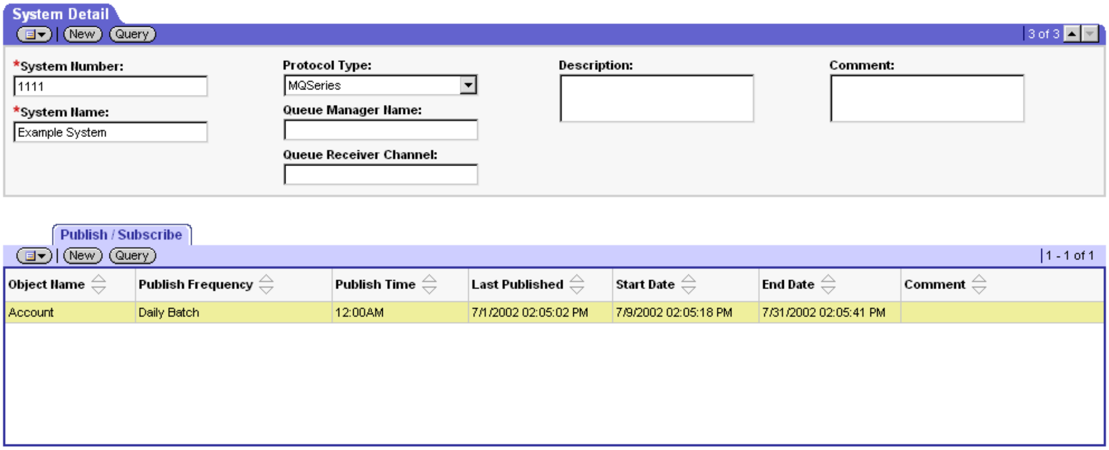
NOTE: The Actual Start value indicates that after you submit the service request, the workflow is triggered immediately based on UTC time.

d In the Component Request Parameters form, click the menu button and then New Record.

e Enter the following information in the parameter columns:

Field	Value
Name	Workflow Process Name
Value	The name of the workflow you used for dailybatch publish, such as CIF Account Dailybatch Publish Workflow

Figure 13 displays the definition of batch-time publish and subscribe capabilities for the example system.



*System Number: 1111

*System Name: Example System

Protocol Type: MQSeries

Queue Manager Name:

Queue Receiver Channel:

Description:

Comment:

Object Name	Publish Frequency	Publish Time	Last Published	Start Date	End Date	Comment
Account	Daily Batch	12:00AM	7/1/2002 02:05:02 PM	7/9/2002 02:05:18 PM	7/31/2002 02:05:41 PM	

Figure 13. Defining system batch-time publish and subscribe capabilities

Key Map

Cross-referencing of customer identification allows organizations to store the Customer Identification from external systems within the Siebel Universal Customer Master Application. This cross-reference permits a one-to-one mapping of customer records across multiple systems throughout the organization. Siebel UCM Applications support Account, Contact, and Household party object cross-referencing.

When external systems send a request message to the Siebel Server with a request to insert a party record or a request to update a party record, the reference records are systematically entered into the Siebel UCM Applications. The Unique Customer Identification number from external systems is embedded within the <Id> tag of the request message.

The System Administrator can manually create reference records from the Siebel UCM Applications Administration Screen.

To create a new Account Reference record

1. Navigate to the Customer Information File Administration Screen.
2. From the Show drop-down list, select References.

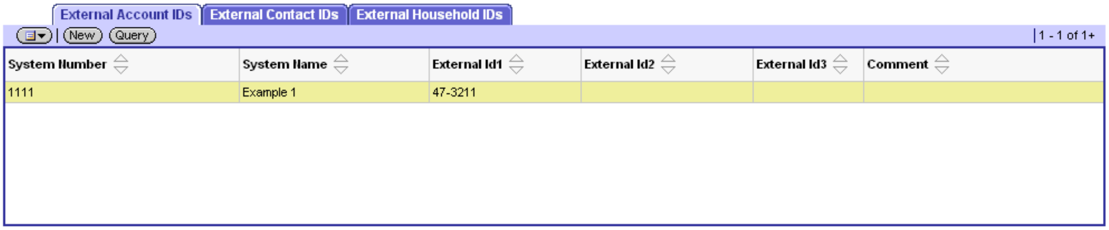
   The Account List, Account Form, and External Account IDs appear.
3. In the Account List, select the record of interest.
4. In the External Account Id view, click the menu button and then New Record.

5 Enter the appropriate information in the available fields to define the external customer and the Unique Account Identification, for each Account party record in Siebel UCM Applications. Use Table 24 for information on each of the fields.

Table 24. Reference Record Fields

Field	Description
System Number	The System Number represents an identifier for an application that accesses Siebel UCM Applications. This number is picked from the Systems entered in the System Registration View.
System Name	The System name is the name of the application that accesses Siebel Universal Customer Master Application. This value is defaulted when a System Number is picked.
External ID1	The External Id1 stores the Unique Identification of the external system record. This value is the only field store in the <Id> tag in request message.
External ID2	The External Id2 stores the second Unique Identification of the external system record.
External ID3	The External Id3 stores the third Unique Identification of the external system record.
Comment	Comments regarding the external system record referencing the record in the Siebel Universal Customer Master Application.
System Number	The System Number represents an identifier for an application that accesses Siebel UCM Applications. This number is picked from the Systems entered in the System Registration View.

Figure 14 displays the creation of an example Account reference record.



System Number	System Name	External Id1	External Id2	External Id3	Comment
1111	Example 1	47-3211			

Figure 14. Creation of an example Account reference record

To create a new Contact Reference record

1. Navigate to the Customer Information File Administration Screen.
2. From the Show drop-down list, select References.
3. Select the External Contact Id tab.

   The Contact List, Contact Form, and External Contact Id appear.
4. In the Contact List Applet, select the record of interest.
5. In the External Contact Id list, click the menu button and then New Record.
6. Enter the appropriate information in the available fields to define the external customer and the Unique Contact Identification, for each Contact party record in the Siebel Universal Customer Master Application.

   Use Table 24 on page 72 for information on each of these fields.

To create a new Household Reference record

1. Navigate to the Customer Information File Administration Screen.
2. From the Show drop-down list, select References.
3. Select the External Household Id tab.

   The Household List, Household Form, and External Household Id appear.
4. In the Household List, select the record of interest.
5. In the External Household Id, click the menu button and then New Record.
6. Enter the appropriate information in the available fields to define the external customer and the Unique Household Identification, for each Household party record in the Siebel Universal Customer Master Application.

   Use Table 24 on page 72 for information on each of these fields

Integration Objects

There are three types of Siebel UCM Applications integration objects:

- "CIF Integration Objects"
- "CIF Envelope Integration Objects"
- "CIF Dispatcher Map Integration Objects"

CIF Integration Objects

The CIF integration object has the hierarchy of Siebel business objects and business components as well as the fields. It maps to the objects in the Siebel eBusiness Application and is created using Integration Object Builder in Siebel Tools; see *Siebel Tools Reference* for further information. This internal integration object is required in order to facilitate the FINS CIF Transaction Manager to package the data gathered. This service gathers the data needed for a particular message and packages it into the hierarchy defined in the internal integration object.

To modify CIF Integration Objects

1. Determine whether or not you need to customize any integration objects listed in Table 30 on page 86, and use Siebel Tools to select the Integration Object tab.
2. Query for name of the integration objects that require modification.
3. Select the components to modify in your integration object and enter the proper information for each of the applicable columns.

NOTE: Only modify the CIF integration objects by activating or deactivating integration component fields and integration components. Large amounts of customization are difficult to upgrade. For a lists of shipped integration objects, see Chapter 5, "Siebel UCM Applications Messages."

CIF Envelope Integration Objects

The envelope integration object is required for all integration business processes. If your integration process requires a slightly different envelope integration object, customize it for your needs using the following procedures. Otherwise use the envelope integration object named CIF Envelope.

To modify CIF envelope integration objects

1. Determine whether or not you need a customized envelope integration object and use Siebel Tools to select the Integration Object tab.
2. Query for name = CIF Envelope.
3. Select the components to modify in your integration object and enter the proper information for each of the applicable columns.
4. After configuring the envelope integration object, compile it to the Siebel Repository File (.srf) file.

NOTE: If you have created a new envelope integration object, make sure to reflect the new configuration when configuring the Siebel Connector components for the Siebel Universal Customer Master Application. Only configure a new envelope integration object if additional integration component fields are required. For details, "CRMML Standard" on page 37 and "About CRMML Envelope and Header Information" on page 90.

CIF Dispatcher Map Integration Objects

The CIF dispatcher map is an integration object that contains the rule sets used by the FINS CIF Dispatcher. The default CIF dispatcher map is CIFDispMap. Siebel UCM Applications use the following predefined dispatcher maps:

- CIFDispMap—Default dispatcher map used
- CIFDispMapAccount—Dispatcher map containing only CIFAccount messages
- CIFDispMapContact—Dispatcher map containing only CIFContact messages

- CIFDispMapHHMisc—Dispatcher map containing the CIFHousehold, CIFFINCORP Account, and CIFPersonal Address messages
- CIFDispMapPerf—Dispatcher map containing a subset of CIFContact messages for improving performance

Deactivate business processes that you are not supporting in a production environment for performance reasons. You can do so by deactivating the user property name corresponding to the business processes that are not applicable.

To view the dispatcher map user properties

1. From Siebel Tools, choose Object Explorer > Integration Object. For more information on using Siebel Tools, see *Siebel Tools Reference*.
2. Query for the dispatcher map name. For example, CIFDispMap.
3. Navigate to the user properties of the dispatcher map to see its user properties.

The name of the user property represents the rule the dispatcher tries to match. An example name follows:

```
CRMML/CIFPartyPackage/DeleteAccount
```

The dispatcher tries to locate the user property name in the message received. If it finds the match, it then uses the information in the value column to determine the values it needs to insert. The value for the previous example name follows:

```
CRMML/CIFPartyPackage/
DeleteAccount;a;b;CIFAccountDelete;CIFAccountDeleteRs;IXMLOperat
ion_DELETE
```

Each value is made up of six tokens that are separated by a semi-colon (;), and each token represents a specific information as outlined in the following table:

Table 25. Dispatcher Map Value User Property

Token	Description	Example
1st	The location to insert the remaining five tokens at runtime	`CRMML/CIFPartyPackage/DeleteAccount`
2nd	System reserved token	`a`
3rd	System reserved token	`b`

Table 25. Dispatcher Map Value User Property

Token	Description	Example
4th	The request integration object	`CIFAccountDelete`
5th	The response integration object	`CIFAccountDeleteRs`
6th	The operation corresponding to <DeleteContact> business process, which is configured in the user property of the FINS CIF Transaction Manager as key to operation	`IXMLOperation_DELETE`

The name column of the map user property must be unique and represents the business process name you want to support. You need to modify the dispatcher map entries to reflect the new business process name. The same principle applies to all the tokens.

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

Siebel UCM Applications Integration Services

Siebel UCM Applications provide an object called a business service, which you can reuse in multiple applications. The Siebel Connector for the Customer Information File provides the following prebuilt business services, which you can configure to meet your business requirements:

- FINS CIF Transaction Manager
- FINS CIF Converter
- FINS CIF Dispatcher
- FINS CIF Security Manager
- FINS CIF Publish/Subscribe Service

You can configure business services by manipulating their user properties, or you can create your own business service in Siebel Tools. For information on Siebel Tools, see *Siebel Tools Reference*.

NOTE: After you have configured your business services to accomplish the tasks required for your business scenario, you need to compile the business service to include the new information in your Siebel repository file (.srf). Then copy the .srf file to the directory where your servers can access it.

FINS CIF Transaction Manager

You can extend this business service by manipulating its Operation user property. The basic format for value entry for each operation is as follows:

CIFOperation_Query

Service/Method/Argument;Argument; or /Method/Argument;Argument;

- Service, Method, and Argument are separated by a forward slash (/).
- Each Argument ends with a semi-colon (;).

CIFOperation_XMLQuery

EAI Siebel Adapter/Query/#XMLHierarchy;

- The default Service name is EAI Siebel Adapter and the default argument name is SiebelMessage.

- SiebelMessage indicates turning off the SiebelMessage.
- XMLHierarchy indicates replacing SiebelMessage with XMLHierarchy.

CIFOperation_GetValue

FINS Industry/BC Facility Service/HierarchySearchSpec/!SiebelMessage;A = > B;

- A = > B means getting argument value of A from argument value of B where argument B is Connector Integration Object Instance's arguments

Table 26 lists the user properties examples for the FINS CIF Transaction Manager.

Table 26. User Properties for FINS CIF Transaction Manager

Operation Name	Value
IXMLOperation_ADD	EAI Siebel Adapter/Insert/
IXMLOperation_DELETE	EAI Siebel Adapter/Delete/
IXMLOperation_QUERY	EAI Siebel Adapter/Query/
IXMLOperation_QUERY_BY_ID	EAI Siebel Adapter/Query/ PrimaryRowId;!SiebelMessage;
IXMLOperation_SYNCH	EAI Siebel Adapter/Synchronize
IXMLOperation_UPDATE	EAI Siebel Adapter/Update

FINS CIF Converter

Table 27 displays the only user property you can configure for this business service. This value appears in the preheader section of your CRMML message.

Table 27. User Property for FINS CIF Converter

Name	Value
XMLEnvIntObjectName	Name of the envelope integration object shipped or customized by you.

The FINS CIF Converter uses the hierarchy represented in the CIF integration object to guide the message through the converting process. If there are elements in the integration object instance received that do not have a definition defined in the integration object definition, the converter errors out. If you expect such a situation, you can set the Ignore Undefined XML Tag parameter on the user property of the corresponding integration object.

NOTE: This user property is created by the wizard and is set to Y. You can turn it off if you wish the converter to error out.

FINS CIF Dispatcher

You can modify both user properties for this business service as shown in Table 28.

Table 28. User Properties for FINS CIF Dispatcher

Name	Value
DispatcherMapName	Name of the dispatcher map shipped or customized by you.
XMLEnvIntObjectName	Name of the envelope integration object shipped or customized by you.

FINS CIF Security Manager

You can modify or extend this business service by manipulating its user properties. The Name column stands for the CIF Transaction Manager Operation Name, while the Value column represents the Field Name for different types of privileges in the System Privileges View. Table 29 displays these user properties.

Table 29. User Properties for FINS CIF Security Manager

Name	Value
IXMLOperation_ADD	Insert
IXMLOperation_DELETE	Delete
IXMLOperation_QUERY	Query
IXMLOperation_UPDATE	Update

Inbound Server Administration

A server task is an instantiation of a server component. To run a server task, you need to run a component request, which requests one or more server tasks to run. The server task you need to run for the Siebel Connector for Universal Customer Master Applications depends on the registered system protocol type, either the Siebel EAI MQSeries Server Transport or the Siebel HTTP Server Transport.

The Siebel EAI MQSeries Server Transport is designed to provide a messaging solution to help you integrate data between Siebel eBusiness Applications and external applications that can interface with the IBM MQSeries. The EAI MQSeries Server Transport transports messages to and from IBM MQSeries queues. In order to run this server task successfully you need to first configure two named subsystems.

To configure named subsystem

1. Navigate to the Server Administration screen, Enterprise Configuration view.
2. Click the Enterprise Profile Configuration view tab.
3. Create a two new records in the Component Profiles list and provide the following information.

	Record 1	**Record 2**
Name	Any name, CIFMQConnSubsy for example	Any name, CIFMQDataSubsys for example
Type	MQSeriesServerSubsys	EAITransportDataHandlingSubsys

NOTE: The subsystem type that you select should have a checkmark in the Is Named Enabled field.

4 For each record, modify the following parameters in the Enterprise Profile Configuration list:

Parameter Name	Record 1	Record 2
MqPhysicalQueueName	Queue name to receive inbound request message from	N/A
MqQueueManagerName	Queue manager name who own the queues	N/A
MqRespPhysicalQueueName	Queue name to send response message to	N/A
MqSleepTime	100 (or longer if required)	N/A
DispatchWorkflowProcess	N/A	CIF Server Party Package Workflow (MQSeries Server Transport Error Handling)

5 Save both records.

For details on this procedure and the Named Subsystem, see *Siebel Server Administration Guide*.

After creating and configuring your Named Subsystem, you need to configure the MQSeries Receiver. For details on configuring MQSeries, see *Transports and Interfaces: Siebel eBusiness Application Integration Volume III*.

To configure MQSeries Receiver parameters

1 Navigate to the Server Administration screen, Server Component view.

2 Query for MQSeries Server Receiver and click the Component Parameters tab.

3 Set the following parameters as follows:

- Receiver Connection Subsystem—Subsystem name created in "To configure named subsystem" on page 81. For example, CIFMQConnSubsys or CIFMQDataSubsys.

- Receiver Method Name—ReceiveDispatch or ReceiveDispatchSend.
- Default Tasks—1 or number of tasks desired.

4. Restart the Siebel Server and make sure the MQSeries Receiver server component is running.

If the registered system is using the HTTP protocol type, you need to configure the Inbound EAI HTTP Transport. The Inbound EAI HTTP Transport is built into the Siebel Web Engine (SWE). To use it, you need to specify certain parameters listed as follows.

To configure the Siebel Web Engine to run the Inbound HTTP Transport

1. Open eapps_sia.cfg file in the \bin subdirectory in the install directory.
2. Locate the section [/cif_%language%], for example [/cif_enu].
3. Add the EnableExtServiceOnly configuration parameter or set it as follows, if it already exists:

   ```
   [/cif_enu]

   ConnectString = <Connect String>

   EnableExtServiceOnly = TRUE
   ```

4. Save and close the configuration file.

After creating and configuring the SWE, configure the required Named Subsystem.

To configure named subsystem

1. Navigate to the Server Administration screen, Enterprise Configuration view.
2. Click the Enterprise Profile Configuration view tab.
3. Create a new record in the Component Profiles list and provide the following information.

   - Name—Name of the Named Subsystem. For example, CIFInboundHTTPDispatch.
   - Type—Type of the Named Subsystem, EAITransportDataHandlingSubsys.

   NOTE: The subsystem type that you select should have a checkmark in the Is Named Enabled field.

4 Modify the following parameter in the Enterprise Profile Configuration list:

DispatchWorkflowProcess—CIF Server Party Package Workflow (MQSeries Server Transport Error Handling)

5 Save the record.

For details on this procedure and the Named Subsystem, see *Siebel Server Administration Guide*.

After creating and configuring your Named Subsystem, configure the Siebel UCM Applications.

To configure the Siebel Universal Customer Master Application to run the Extension Service

1 Open cif.cfg file in the \bin subdirectory in the install directory.

2 Locate the section `[/HTTP Services]`, or add this section if it is not found.

3 Add the services and corresponding named subsystems configured. For example:

```
[/HTTP Services]

SiebelCIFContact = CIFInboundHTTPDispatch
```

The name SiebelCIFContact in the queryString matches the name CIFInboundHTTPDispatch, which in turn looks up for the named subsystem list and dispatches accordingly.

4 Save and close the configuration file.

NOTE: For details on creating and configuring server tasks, see *Siebel Server Administration Guide*, and for details on configuring MQSeries and HTTP, see *Transports and Interfaces: Siebel eBusiness Application Integration Volume III*.

Siebel UCM Applications Messages 5

Siebel Universal Customer Master Applications (Siebel UCM Applications) provide application services interfaces to interact and manage the customer information stored in the Customer Master.

NOTE: This chapter uses former naming conventions for the Siebel UCM Applications product when referencing hard-coded product interface messages. References to CIF (Customer Information File) and FINS CIF (Financial Services Customer Information File) are valid for the Siebel UCM Applications for Industry Applications product.

About Siebel Universal Customer Master Messages

The application service interfaces supported with Siebel UCM Applications are:

- **Manage Customer**—inbound message allowing querying, inserting, updating, and deleting of Contact, Account, and Household information, including relationships.
- **Manage Address**—inbound message allowing querying, inserting, updating, and deleting of Addresses for Contacts, Accounts, and Households.
- **Manage Profile**—inbound message allowing querying, inserting, updating, and deleting of Contact, Account, and Household profile information.
- **Manage Activities**—inbound message allowing querying, inserting, updating, and deleting of Contact, Account, and Household activities.
- **Manage Products**—inbound message allowing querying, inserting, updating, and deleting of Products associated with Contacts, Accounts, and Households.
- **Lookup Customer by Product**—inbound message allowing querying Contacts, Accounts, and Households based on a Product Identification number.
- **Lookup Relationships**—inbound message allowing querying Relationships between Contacts, Accounts, and Households.
- **Lookup Batch Address**—inbound message allowing querying of Contacts, Accounts, and Households based on Address criteria.
- **Publish Customer**—outbound message used for publishing.

Table 30 on page 86 lists out each of the messages from Siebel UCM Applications specifications.

Table 30. Siebel Universal Customer Master Application Messages

Integration Object Name	Business Processes/Operations
CIFAccount	LookUpAccount
CIFAccountActivity	LookUpAccountActivity
CIFAccountActivityDelete	DeleteAccountActivity

Table 30. Siebel Universal Customer Master Application Messages

Integration Object Name	Business Processes/Operations
CIFAccountActivityInsert	InsertAccountActivity
CIFAccountActivityUpdate	UpdateAccountActivity
CIFAccountAddress	LookUpAccountAddress
CIFAccountAddressDelete	DeleteAccountAddress
CIFAccountAddressInsert	InsertAccountAddress
CIFAccountAddressUpdate	UpdateAccountAddress
CIFAccountDelete	DeleteAccount
CIFAccountInsert	InsertAccount
CIFAccountProduct	LookUpAccountProduct
CIFAccountProductDelete	DeleteAccountProduct
CIFAccountProductInsert	InsertAccountProduct
CIFAccountProductUpdate	UpdateAccountProduct
CIFAccountProfile	LookUpAccountProfile
CIFAccountProfileByName	LookUpAccountProfileByName
CIFAccountProfileDelete	DeleteAccountProfile
CIFAccountProfileInsert	InsertAccountProfile
CIFAccountProfileUpdate	UpdateAccountProfile
CIFAccountRelationship	LookUpAccountRelationship
CIFAccountRelationshipDelete	DeleteAccountRelationship
CIFAccountRelationshipInsert	InsertAccountRelationship
CIFAccountRelationshipUpdate	UpdateAccountRelationship
CIFAccountUpdate	UpdateAccount
CIFContact	LookUpContact
CIFContactActivity	LookUpContactActivity

Table 30. Siebel Universal Customer Master Application Messages

Integration Object Name	Business Processes/Operations
CIFContactActivityByName	LookUpContactActivityByName
CIFContactActivityDelete	DeleteContactActivity
CIFContactActivityInsert	InsertContactActivity
CIFContactActivityUpdate	UpdateContactActivity
CIFContactAddress	LookUpContactAddress
CIFContactAddressByName	LookUpContactAddressByName
CIFContactAddressDelete	DeleteContactAddress
CIFContactAddressInsert	InsertContactAddress
CIFContactAddressUpdate	UpdateContactAddress
CIFContactByName	LookUpContactByName
CIFContactDelete	DeleteContact
CIFContactExternalId	LookUpContactExternalId
CIFContactInsert	InsertContact
CIFContactProduct	LookUpContactProduct
CIFContactProductByName	LookUpContactProductByName
CIFContactProductDelete	DeleteContactProduct
CIFContactProductInsert	InsertContactProduct
CIFContactProductUpdate	UpdateContactProduct
CIFContactProfile	LookUpContactProfile
CIFContactProfileByName	LookUpContactProfileByName
CIFContactProfileDelete	DeleteContactProfile
CIFContactProfileInsert	InsertContactProfile
CIFContactProfileUpdate	UpdateContactProfile
CIFContactRelationship	LookUpContactRelationship

Table 30. Siebel Universal Customer Master Application Messages

Integration Object Name	Business Processes/Operations
CIFContactRelationshipByName	LookUpContactRelationshipByName
CIFContactRelationshipDelete	DeleteContactRelationship
CIFContactRelationshipInsert	InsertContactRelationship
CIFContactRelationshipUpdate	UpdateContactRelationship
CIFContactUpdate	UpdateContact
CIFFINCORPAccount	LookUpFINCORPAccount
CIFFINCORPAccountByName	LookUpFINCORPAccountByName
CIFHousehold	LookUpHousehold
CIFHouseholdByName	LookUpHouseholdByName
CIFHouseholdDelete	DeleteHousehold
CIFHouseholdInsert	InsertHousehold
CIFHouseholdUpdate	UpdateHousehold
CIFPersonalAddress	LookUpPersonalAddress
CIFPersonalAddressByState	LookUpPersonalAddressByState

About CRMML Envelope and Header Information

To support the key features of Siebel UCM Applications such as system security check, privilege checking function, and so on, the Siebel Connector for Universal Customer Master Applications is designed to generate the CRMML Message Header. An example of the header is shown below:

```
<! CRMML Header section starts>
      <CRMHeader>
            <FromSystem>
                  <SystemInfo type="ID">
                  <SystemId>www.mycompany.com</SystemId>
                  <SystemRole>Insurance Application</SystemRole>
                  <SystemName>Client UI System</SystemName>
                  </SystemInfo>
            </FromSystem>
            <ToSystem>
                  <SystemInfo type="ID">
                  <SystemId>123-456-789</SystemId>
                  <SystemRole>Service Provider</SystemRole>
                  <SystemName>Siebel CIF System</SystemName>
                  </SystemInfo>
            </ToSystem>
            <MessageDescriptor>
                  <MessageId>11SC3I2002053116:43:45</MessageId>
                  <TransactionScope>any</TransactionScope>
                  <Timestamp>01/01/2001</Timestamp>
                  <TimeToLive>365</TimeToLive>
            </MessageDescriptor>
</CRMHeader>
```

Certain values in the header are automatically generated by the connector, but you can specify others in the envelope integration object. Table 31 describes the elements and how they are configured. For system generated attributes, these is no configuration available.

Table 31. Header Elements

Element	Structure	Attribute	Description	Default	Configuration
CRMML	(top level)				
CRMHeader	In CRMML		Header aggregate		
FromSystem	In CRMHeader		FromSystem aggregate		
SystemInfo	In FromSystem		SystemInfo aggregate		
		Type	Describes the type of SystemID such as ID or URL	ID	CIF Envelope Integration Object- > SystemInfo Integration Component- > type Integration Component Field, XML Literal Value
		SystemId	Identifier of the system	123-456-789	CIF Envelope Integration Object- > SystemInfo Integration Component- > SystemId Integration Component Field, XML Literal Value
		SystemName	Name of the system	SiebelCIFSyste m	CIF Envelope Integration Object- > SystemInfo Integration Component- > SystemName Integration Component Field, XML Literal Value
		SystemRole	Role the system has	ServiceProvider	CIF Envelope Integration Object- > SystemInfo Integration Component- > SystemRole Integration Component Field, XML Literal Value
ToSystem	In CRMHeader		ToSystem aggregate		
SystemInfo	In ToSystem		SystemInfo aggregate		

Table 31. Header Elements

Element	Structure	Attribute	Description	Default	Configuration
		Type	Describes the type of SystemID such as ID or URL	ID	CIF Envelope Integration Object- > SystemInfo Integration Component- > ToSystem_type Integration Component Field, XML Literal Value
		SystemId	Identifier of the system		System Generated
		SystemName	Name of the system		System Generated
		SystemRole	Role the system has		System Generated
MessageDescripto	In CRMHeader				
		MessageId	Identifier of the message		System Generated
		TransactionScope	Scope of the transaction. Currently, only support any		CIF Envelope Integration Object- > MessageDescriptor Integration Component- > MessageDescriptor_Tr asactionScope Integration Component Field, XML Literal Value
		Timestamp	Time stamp of when the message is constructed		System Generated
		TimeToLive	Expiration length of the message from Timestamp value		CIF Envelope Integration Object- > MessageDescriptor Integration Component- > MessageDescriptor_T oLive Integration Component Field, XML Literal Value

Universally Unique Identifier (UUID)

The Common Key Infrastructure provides functionality to create a universally unique identifier (UUID) for each new record created in the Siebel Universal Customer Master Application. The key is unique across multiple Siebel Applications as well as other applications connected to the Siebel Universal Customer Master Application.

Siebel Universal Customer Master Application Messages

The Siebel Universal Customer Master Application messages provide a solution for enterprise customer, partner, and prospect data management and access. The Siebel Universal Customer Master Application messages, embedded in Customer Relationship Management Markup Language (CRMML) XML messages, interact with systems across the enterprise application network and function as the system of record for customer data.

Contact Profile

CIF Contact Profile messages are used in insert, query, update, and delete operations to maintain Contact Profile Information. LookUpContact Profile and LookUpContactProfileByName are used for query operations. UpdateContactProfile, InsertContactProfile, and DeleteContactProfile are message pairs for request and response operations. Figure 15 describes the structure of this message.

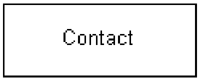


Figure 15. CIF Contact Profile Structure

Table 32. Contact Profile Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfile	Contact	LookUpContactProfile	1:PartyUId 2: Id
CIFContactProfileRs	Contact	LookUpContactProfileResponse	1:PartyUId 2: Id

Table 33. Contact Profile Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfileByName	Contact	LookUpContactProfileByName	1:PartyUId 2: Id
CIFContactProfileByNameRs	Contact	LookUpContactProfileByNameResponse	1:PartyUId 2: Id

Table 34. Contact Profile Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfileUpdate	Contact	UpdateContactProfile	1:PartyUId 2: Id
CIFContactProfileUpdateRs	Contact	UpdateContactProfileResponse	1:PartyUId 2: Id

Table 35. Contact Profile Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfileInsert	Contact	InsertContactProfile	1:PartyUId 2: Id
CIFContactProfileUpdateRs	Contact	InsertContactProfileResponse	1:PartyUId 2: Id

Table 36. Contact Profile Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfileDelete	Contact	DeleteContactProfile	1:PartyUId 2: Id
CIFContactProfileDeleteRs	Contact	DeleteContactProfileResponse	1:PartyUId 2: Id

Table 37. Contact Profile Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Alias	Alias	DTYPE_TEXT	50		
Assistant	Assistant	DTYPE_TEXT	50		
Assistant Phone #	AssistantPhone	DTYPE_TEXT	40		
Best Call Time	BestCallTime	DTYPE_TEXT	30		Bound by FINCORP_CONTACT_CALL_TIME
Cellular Phone #	CellularPhone	DTYPE_TEXT	40		
Customer Value	CustomerValue	DTYPE_TEXT	30		Bound by FIN_CON_VALUE_TYPE
Date of Birth	DateofBirth	DTYPE_DATE	7		
Email Address	EmailAddress	DTYPE_TEXT	50		
Email Address 2	EmailAddress2	DTYPE_TEXT	50		
Fax Phone #	FaxPhone	DTYPE_TEXT	40		
First Name	FirstName	DTYPE_TEXT	50	Y	
Home Phone #	HomePhone	DTYPE_TEXT	40		

Table 37. Contact Profile Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Id	Id	DTYPE_ID	15	Y	This tag is required for Id cross-referencing. Tag stores incoming external Id.
Integration Id	IntegrationId	DTYPE_TEXT	30		
Job Title	JobTitle	DTYPE_TEXT	75		
Last Name	LastName	DTYPE_TEXT	50	Y	
M/F	MF	DTYPE_TEXT	30		Bound by SEX_MF
M/M	MM	DTYPE_TEXT	15		Bound by MR_MS
Marital Status	MaritalStatus	DTYPE_TEXT	30		Bound by MARITAL_STATUS
Middle Name	MiddleName	DTYPE_TEXT	50		
Mother Maiden Name	MotherMaidenName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100		
Person UId	PersonUId	DTYPE_TEXT	100		
Personal Contact	PersonalContact	DTYPE_TEXT	1		
Preferred Communications	PreferredCommunications	DTYPE_TEXT	30		Bound by OFFER_MEDIA
Privacy Code	PrivacyCode	DTYPE_TEXT	30		Bound by PRIVACY_CODE
Social Security Number	SocialSecurityNumber	DTYPE_TEXT	20		
Status	Status	DTYPE_TEXT	30		Bound by CONTACT_STATUS
Suppress All Calls	SuppressAllCalls	DTYPE_TEXT	1		
Suppress All Emails	SuppressAllEmails	DTYPE_TEXT	1		

Table 37. Contact Profile Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Suppress All Faxes	SuppressAllFaxes	DTYPE_TEXT	1		
Suppress All Mailings	SuppressAllMailings	DTYPE_TEXT	1		
Suppress Share	SuppressShare	DTYPE_TEXT	1		
Time Zone Id	TimeZoneId	DTYPE_ID	15		
Time Zone Name	TimeZoneName	DTYPE_TEXT	100		Used for picking a related entity
Work Phone #	WorkPhone	DTYPE_TEXT	40		
Work Phone Extension	WorkPhoneExtension	DTYPE_TEXT	10		
operation	operation	DTYPE_TEXT	30		System field

Contact Address

CIF Contact Address messages are used in insert, query, update, and delete operations to maintain Contact Address Information. LookUpContact Address and LookUpContactAddressByName are used for query operations. UpdateContactAddress, InsertContactAddresse, and DeleteContactAddress are message pairs for request and response operations. Figure 16 describes the structure of this message.

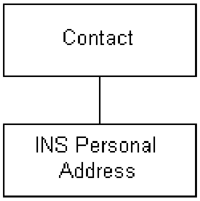


Figure 16. CIF Contact Address Structure

Table 38. Contact Address Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddress	Contact	LookUpContactAddress	1:PartyUId 2: Id
CIFContactAddressRs	Contact	LookUpContactAddressResponse	1:PartyUId 2: Id

Table 39. Contact Address Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddressByName	Contact	LookUpContactAddressByName	1:PartyUId 2: Id
CIFContactAddressByNameRs	Contact	LookUpContactAddressByNameResponse	1:PartyUId 2: Id

Table 40. Contact Address Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddressUpdate	Contact	UpdateContactAddress	1:PartyUId 2: Id
CIFContactAddressUpdateRs	Contact	UpdateContactAddressResponse	1:PartyUId 2: Id

Table 41. Contact Address Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddressInsert	Contact	InsertContactAddress	1:PartyUId 2: Id
CIFContactAddressInsertRs	Contact	InsertContactAddressResponse	1:PartyUId 2: Id

Table 42. Contact Address Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddressDelete	Contact	DeleteContactAddress	1:PartyUId 2: Id
CIFContactAddressDeleteRs	Contact	DeleteContactAddressResponse	1:PartyUId 2: Id

Table 43. Contact Address Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Contact					
Fax Phone #	FaxPhone	DTYPE_TEXT	40		
First Name	FirstName	DTYPE_TEXT	50	Y	
Id	Id	DTYPE_ID	15	Y	This tag is required for Id cross-referencing. Tag stores incoming external Id.
Integration Id	IntegrationId	DTYPE_TEXT	30		

Table 43. Contact Address Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Job Title	JobTitle	DTYPE_TEXT	75		
Last Name	LastName	DTYPE_TEXT	50	Y	
M/M	MM	DTYPE_TEXT	15		Bound by MR_MS
Middle Name	MiddleName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100		
Person UId	PersonUId	DTYPE_TEXT	100		
Privacy Code	PrivacyCode	DTYPE_TEXT	30		
Status	Status	DTYPE_TEXT	30		Bound by CONTACT_ST ATUS
Work Phone #	WorkPhone	DTYPE_TEXT	40		
Work Phone Extension	WorkPhoneExtension	DTYPE_TEXT	10		
operation	operation	DTYPE_TEXT	30		
Contact_INS Personal Address					
INS Personal Address Id	INSPersonalAddressId	DTYPE_ID	15		
INS Personal Address Name	INSPersonalAddressName	DTYPE_TEXT	100		
INS Personal Address Type	INSPersonalAddressType	DTYPE_TEXT	30		
INS Personal City	INSPersonalCity	DTYPE_TEXT	50	Y	
INS Personal Country	INSPersonalCountry	DTYPE_TEXT	30		
INS Personal County	INSPersonalCounty	DTYPE_TEXT	50		

Table 43. Contact Address Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Job Title	JobTitle	DTYPE_TEXT	75		
Last Name	LastName	DTYPE_TEXT	50	Y	
M/M	MM	DTYPE_TEXT	15		Bound by MR_MS
Middle Name	MiddleName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100		
Person UId	PersonUId	DTYPE_TEXT	100		
Privacy Code	PrivacyCode	DTYPE_TEXT	30		
Status	Status	DTYPE_TEXT	30		Bound by CONTACT_ST ATUS
Work Phone #	WorkPhone	DTYPE_TEXT	40		
Work Phone Extension	WorkPhoneExtension	DTYPE_TEXT	10		
operation	operation	DTYPE_TEXT	30		
Contact_INS Personal Address					
INS Personal Address Id	INSPersonalAddressId	DTYPE_ID	15		
INS Personal Address Name	INSPersonalAddressName	DTYPE_TEXT	100		
INS Personal Address Type	INSPersonalAddressType	DTYPE_TEXT	30		
INS Personal City	INSPersonalCity	DTYPE_TEXT	50	Y	
INS Personal Country	INSPersonalCountry	DTYPE_TEXT	30		
INS Personal County	INSPersonalCounty	DTYPE_TEXT	50		

Table 43. Contact Address Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
INS Personal Integration Id	INSPersonalIntegrationId	DTYPE_TEXT	30		
INS Personal Postal Code	INSPersonalPostalCode	DTYPE_TEXT	30		
INS Personal State	INSPersonalState	DTYPE_TEXT	10		
INS Personal Street Address	INSPersonalStreetAddress	DTYPE_TEXT	200	Y	
INS Personal Street Address 2	INSPersonalStreetAddress2	DTYPE_TEXT	100		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
operation	operation	DTYPE_TEXT	30		

Contact Activity

CIF Contact Activity messages are used in insert, query, update, and delete operations to maintain Contact Activity Information. LookUpContactActivity and LookUpContactActivityByName are used for query operations. UpdateContactActivity, InsertContactActivity, and DeleteContactActivity are message pairs for request and response operations. Figure 17 describes the structure of this message.

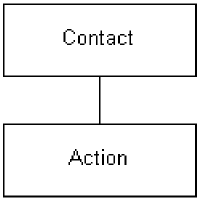


Figure 17. CIF Contact Activity Structure

Table 44. Contact Activity Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactActivity	Contact	LookUpContactActivity	1:PartyUId 2: Id
CIFContactActivityRs	Contact	LookUpContactActivityResponse	1:PartyUId 2: Id

Table 45. Contact Activity Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactActivityByName	Contact	LookUpContactActivityByName	1:PartyUId 2: Id
CIFContactActivityByNameRs	Contact	LookUpContactActivityByNameResponse	1:PartyUId 2: Id

Table 46. Contact Activity Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactActivityUpdate	Contact	UpdateContactActivity	1:PartyUId 2: Id
CIFContactActivityUpdateRs	Contact	UpdateContactActivityResponse	1:PartyUId 2: Id

Table 47. Contact Activity Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactActivityInsert	Contact	InsertContactActivity	1:PartyUId 2: Id
CIFContactActivityInsertRs	Contact	InsertContactActivityResponse	1:PartyUId 2: Id

Table 48. Contact Activity Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactActivityDelete	Contact	DeleteContactActivity	1:PartyUId 2: Id
CIFContactActivityDeleteRs	Contact	DeleteContactActivityResponse	1:PartyUId 2: Id

Table 49. Contact Activity Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Contact					
Assistant	Assistant	DTYPE_TEXT	50		
Assistant Phone #	AssistantPhone	DTYPE_TEXT	40		
Cellular Phone #	CellularPhone	DTYPE_TEXT	40		
Email Address	EmailAddress	DTYPE_TEXT	50		
Fax Phone #	FaxPhone	DTYPE_TEXT	40		
First Name	FirstName	DTYPE_TEXT	50	Y	
Home Phone #	HomePhone	DTYPE_TEXT	40		

Table 49. Contact Activity Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Id	Id	DTYPE_ID	15	Y	This tag is required for Id cross-referencing. Tag stores incoming external Id.
Integration Id	IntegrationId	DTYPE_TEXT	30		
Job Title	JobTitle	DTYPE_TEXT	75		
Last Name	LastName	DTYPE_TEXT	50	Y	
M/F	MF	DTYPE_TEXT	30		Bound by SEX_MF
M/M	MM	DTYPE_TEXT	15		Bound by MR_MS
Middle Name	MiddleName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100		
Person UId	PersonUId	DTYPE_TEXT	100		
Preferred Communications	PreferredCommunications	DTYPE_TEXT	30		Bound by OFFER_MEDIA
Status	Status	DTYPE_TEXT	30		Bound by CONTACT_ST ATUS
Work Phone #	WorkPhone	DTYPE_TEXT	40		
Work Phone Extension	WorkPhoneExtension	DTYPE_TEXT	10		
operation	operation	DTYPE_TEXT	30		
Action					
Account Id	AccountId	DTYPE_ID	15		
Account Location	AccountLocation	DTYPE_TEXT	50		

Table 49. Contact Activity Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Account Name	AccountName	DTYPE_TEXT	100		Used for picking a related entity.
Audience	Audience	DTYPE_TEXT	30		Bound by ACTIVITY_AUDIENCE_TYPE
Category	Category	DTYPE_TEXT	30		Bound by TODO_TYPE
Description	Description	DTYPE_TEXT	100		
Done	Done	DTYPE_DATETIME	7		
Done Flag	DoneFlag	DTYPE_TEXT	1		
Due	Due	DTYPE_DATETIME	7		
Duration Minutes	DurationMinutes	DTYPE_NUMBER	22		Bound by APPOINTMENT_DURATION
Id	Id	DTYPE_ID	15		
Integration Id	IntegrationId	DTYPE_TEXT	30		
MeetingLocation	MeetingLocation	DTYPE_TEXT	100		
No Sooner Than Date	NoSoonerThanDate	DTYPE_UTCDATETIME	7		
Planned	Planned	DTYPE_UTCDATETIME	7		
Planned Completion	PlannedCompletion	DTYPE_UTCDATETIME	7		
Priority	Priority	DTYPE_TEXT	30		
Started	Started	DTYPE_UTCDATETIME	7		

Table 49. Contact Activity Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Status	Status	DTYPE_TEXT	30		Bound by EVENT_STATUS
Type	Type	DTYPE_TEXT	30	Y	Bound by TODO_TYPE
operation	operation	DTYPE_TEXT	30		

Contact Product

CIF Contact Products messages are used in insert, query, update, and delete operations to maintain Contact Product (Financial Account) Information. LookUpContactProduct and LookUpContactProductByName are used for query operations. UpdateContactProduct, InsertContactProduct, and DeleteContactProduct are message pairs for request and response operations. Figure 18 describes the structure of this message

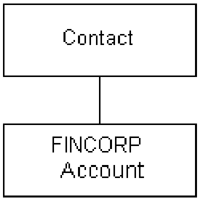


Figure 18. CIF Contact Product Structure

Table 50. Contact Product Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProduct	Contact	LookUpContactProduct	1:PartyUId 2: Id
CIFContactProductRs	Contact	LookUpContactProductResponse	1:PartyUId 2: Id

Table 51. Contact Product Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProductByName	Contact	LookUpContactProductByName	1:PartyUId 2: Id
CIFContactProductByNameR	Contact	LookUpContactProductByNameResponse	1:PartyUId 2: Id

Table 52. Contact Product Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProductUpdate	Contact	UpdateContactProduct	1:PartyUId 2: Id
CIFContactProductUpdateRs	Contact	UpdateContactProductResponse	1:PartyUId 2: Id

Table 53. Contact Product Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProductInsert	Contact	InsertContactProduct	1:PartyUId 2: Id
CIFContactProductInsertRs	Contact	InsertContactProductResponse	1:PartyUId 2: Id

Table 54. Contact Product Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProductDelete	Contact	DeleteContactProduct	1:PartyUId 2: Id
CIFContactProductDeleteRs	Contact	DeleteContactProductResponse	1:PartyUId 2: Id

Table 55. Contact Product Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Contact					
Assistant	Assistant	DTYPE_TEXT	50		
Assistant Phone #	AssistantPhone	DTYPE_TEXT	40		
Cellular Phone #	CellularPhone	DTYPE_TEXT	40		
Email Address	EmailAddress	DTYPE_TEXT	50		
Fax Phone #	FaxPhone	DTYPE_TEXT	40		
First Name	FirstName	DTYPE_TEXT	50	Y	
Home Phone #	HomePhone	DTYPE_TEXT	40		
Id	Id	DTYPE_ID	15	Y	This tag is required for Id cross-referencing. Tag stores incoming external Id.
Integration Id	IntegrationId	DTYPE_TEXT	30		
Job Title	JobTitle	DTYPE_TEXT	75		
Last Name	LastName	DTYPE_TEXT	50	Y	

Table 55. Contact Product Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
M/F	MF	DTYPE_TEXT	255		Bound by SEX_MF
M/M	MM	DTYPE_TEXT	15		Bound by MR_MS
Middle Name	MiddleName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100		
Person UId	PersonUId	DTYPE_TEXT	100		
Preferred Communications	PreferredCommunications	DTYPE_TEXT	30		Bound by OFFER_MEDIA
Status	Status	DTYPE_TEXT	30		
Work Phone #	WorkPhone	DTYPE_TEXT	40		
Work Phone Extension	WorkPhoneExtension	DTYPE_TEXT	10		
operation	operation	DTYPE_TEXT	30		
FINCORP Account					
Account Branch ABA	AccountBranchABA	DTYPE_TEXT	50		Used for picking a related entity.
Account Branch Id	AccountBranchId	DTYPE_ID	15		
Account Holder	AccountHolder	DTYPE_TEXT	100		Used for picking a related entity.
Account Name	AccountName	DTYPE_TEXT	100		
Account Number	AccountNumber	DTYPE_TEXT	100	Y	
Account Status	AccountStatus	DTYPE_TEXT	30		Bound by FINS_CRPST_LFLN_IPFS_PLC_MLOV

Table 55. Contact Product Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Account Type	AccountType	DTYPE_TEXT	30		Bound by FINCORP_PROD_ADMIN_CLASS_MLOV
Acct Gen - Branch Id	AcctGen-BranchId	DTYPE_ID	15		
Available Balance	AvailableBalance	DTYPE_CURRENCY	22		
Branch	Branch	DTYPE_TEXT	50		
Current Balance	CurrentBalance	DTYPE_CURRENCY	22		
Id	Id	DTYPE_ID	15		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Name	Name	DTYPE_TEXT	100	Y	Used for picking a related entity
Organization Id	OrganizationId	DTYPE_ID	15		
Product Id	ProductId	DTYPE_ID	15	Y	
Relationship Limit	RelationshipLimit	DTYPE_CURRENCY	22		
Start Date	StartDate	DTYPE_DATE	7		
operation	operation	DTYPE_TEXT	30		

Contact Relationship

CIF Contact Relationship messages are used in insert, query, update, and delete operations to maintain Contact Relationships. LookUpContactRelationship and LookUpContactRelationshipByName are used for query operations. UpdateContactRelationship, InsertContactRelationship and DeleteContactRelationship are message pairs for request and response operations. Figure 19 describes the structure of this message

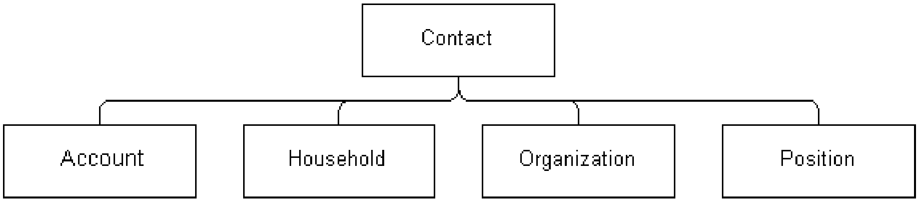


Figure 19. CIF Contact Relationship Structure

Table 56. Contact Relationship Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationship	Contact	LookUpContactRelationship	1:PartyUId 2: Id
CIFContactRelationshipRs	Contact	LookUpContactRelationshipResponse	1:PartyUId 2: Id

Table 57. Contact Relationship Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationshipByName	Contact	LookUpContactRelationshipByName	1:PartyUId 2: Id
CIFContactRelationshipByNameRs	Contact	LookUpContactRelationshipByNameResponse	1:PartyUId 2: Id

Table 58. Contact Relationship Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationshipUpdate	Contact	UpdateContactRelationship	1:PartyUId 2: Id
CIFContactRelationshipUpdateRs	Contact	UpdateContactRelationshipResponse	1:PartyUId 2: Id

Table 59. Contact Relationship Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationshipInsert	Contact	InsertContactRelationship	1:PartyUId 2: Id
CIFContactRelationshipInsertRs	Contact	InsertContactRelationshipResponse	1:PartyUId 2: Id

Table 60. Contact Relationship Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationshipDelete	Contact	DeleteContactRelationship	1:PartyUId 2: Id
CIFContactRelationshipDeleteRs	Contact	DeleteContactRelationshipResponse	1:PartyUId 2: Id

Table 61. Contact Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Contact					
Assistant	Assistant	DTYPE_TEXT	50		
Assistant Phone #	AssistantPhone	DTYPE_TEXT	40		
Cellular Phone #	CellularPhone	DTYPE_TEXT	40		
Email Address	EmailAddress	DTYPE_TEXT	50		
Fax Phone #	FaxPhone	DTYPE_TEXT	40		
First Name	FirstName	DTYPE_TEXT	50	Y	
Home Phone #	HomePhone	DTYPE_TEXT	40		
Id	Id	DTYPE_ID	15	Y	This tag is required for Id cross-referencing. Tag stores incoming external Id.
Integration Id	IntegrationId	DTYPE_TEXT	30		
Job Title	JobTitle	DTYPE_TEXT	75		
Last Name	LastName	DTYPE_TEXT	50	Y	
M/F	MF	DTYPE_TEXT	30		Bound by SEX_MF
M/M	MM	DTYPE_TEXT	15		Bound by MR_MS
Manager First Name	ManagerFirstName	DTYPE_TEXT	50		
Manager Id	ManagerId	DTYPE_ID			
Manager Integration Id	ManagerIntegrationId	DTYPE_TEXT	30		

Table 61. Contact Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Manager Last Name	ManagerLastName	DTYPE_TEXT	50		Used for picking a related entity.
Manager Middle Name	ManagerMiddleName	DTYPE_TEXT	50		
Manager Primary Organization	ManagerPrimaryOrganization	DTYPE_TEXT	100		
Manager Primary Organization Id	ManagerPrimaryOrganizationId	DTYPE_ID	15		
Manager UId	ManagerUId	DTYPE_TEXT	100		
Middle Name	MiddleName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100		
Person UId	PersonUId	DTYPE_TEXT	100		
Preferred Communications	PreferredCommunications	DTYPE_TEXT	30		Bound by OFFER_MEDIA
Status	Status	DTYPE_TEXT	30		Bound by CONTACT_ST ATUS
Work Phone #	WorkPhone	DTYPE_TEXT	40		
Work Phone Extension	WorkPhoneExtension	DTYPE_TEXT	10		
Operation	operation	DTYPE_TEXT	30		System field
Account					
Account	Account	DTYPE_TEXT	100	Y	
Account Location	AccountLocation	DTYPE_TEXT	50		
Account Organization	AccountOrganization	DTYPE_TEXT	100		
Account Party UId	AccountPartyUId	DTYPE_TEXT	100		

Table 61. Contact Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Account Row Id	AccountRowId	DTYPE_ID	15		
Account Status	AccountStatus	DTYPE_TEXT	30		Bound by ACCOUNT_STATUS
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Related Account Integration Id	RelatedAccountIntegrationId	DTYPE_TEXT	30		
Operation	operation	DTYPE_TEXT	30		System field
Household					
Household Id	HouseholdId	DTYPE_ID	30		
Household Integration Id	HouseholdIntegrationId	DTYPE_TEXT	30		
Household Name	HouseholdName	DTYPE_TEXT	50		
Household Number	HouseholdNumber	DTYPE_TEXT	100		
Household Party UId	HouseholdPartyUId	DTYPE_TEXT	100		
Household Status	HouseholdStatus	DTYPE_TEXT	30		Bound by HOUSEHOLD_STATUS
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Operation	operation	DTYPE_TEXT	30		System field
Organization					
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Organization	Organization	DTYPE_TEXT	50		
Organization Id	OrganizationId	DTYPE_ID	30		

Table 61. Contact Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Organization Integration Id	OrganizationIntegrationId	DTYPE_TEXT	30		
Operation	operation	DTYPE_TEXT	30		System field
Position					
Division	Division	DTYPE_TEXT	255		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Position	Position	DTYPE_TEXT	50		
Position Id	PositionId	DTYPE_ID			
Position Integration Id	PositionIntegrationId	DTYPE_TEXT	30		
Sales Rep	SalesRep	DTYPE_TEXT	50		
Operation	operation	DTYPE_TEXT	30		System field

Contact

CIF Contact messages are used in insert, query, update, and delete operations to maintain Contact information. LookUpContact and LookUpContactByName are used for query operations. UpdateContact, InsertContact, and DeleteContact are message pairs for request and response operations. Figure 20 describes the structure of this message.

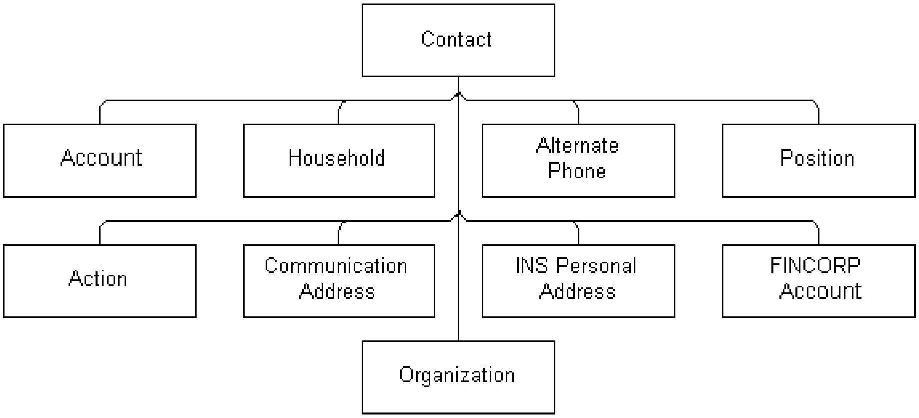


Figure 20. CIF Contact Structure

Table 62. Contact Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContact	Contact	LookUpContact	1:PartyUId 2: Id
CIFContactRs	Contact	LookUpContactResponse	1:PartyUId 2: Id

Table 63. Contact Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactByName	Contact	LookUpContactByName	1:PartyUId 2: Id
CIFContactByNameRs	Contact	LookUpContactByNameResponse	1:PartyUId 2: Id

Table 64. Contact Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactUpdate	Contact	UpdateContact	1:PartyUId 2: Id
CIFContactUpdateRs	Contact	UpdateContactResponse	1:PartyUId 2: Id

Table 65. Contact Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactInsert	Contact	InsertContact	1:PartyUId 2: Id
CIFContactInsertRs	Contact	InsertContactResponse	1:PartyUId 2: Id

Table 66. Contact Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactDelete	Contact	DeleteContact	1:PartyUId 2: Id
CIFContactDeleteRs	Contact	DeleteContactResponse	1:PartyUId 2: Id

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Contact					
Alias	Alias	DTYPE_TEXT	50		
Assistant	Assistant	DTYPE_TEXT	50		
Assistant Phone #	AssistantPhone	DTYPE_TEXT	40		
Best Call Time	BestCallTime	DTYPE_TEXT	30		Bound by FINCORP_CONTACT_CALL_TIME
Cellular Phone #	CellularPhone	DTYPE_TEXT	40		
Customer Value	CustomerValue	DTYPE_TEXT	30		Bound by FIN_CON_VALUE_TYPE
Date of Birth	DateofBirth	DTYPE_DATE	7		
Email Address	EmailAddress	DTYPE_TEXT	50		
Email Address 2	EmailAddress2	DTYPE_TEXT	50		
Fax Phone #	FaxPhone	DTYPE_TEXT	40		
First Name	FirstName	DTYPE_TEXT	50	Y	
Home Phone #	HomePhone	DTYPE_TEXT	40		
Id	Id	DTYPE_ID	15	Y	This tag is required for Id cross-referencing. Tag stores incoming external Id.
Income	Income	DTYPE_CURRENCY	22		

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Income Currency Code	IncomeCurrencyCode	DTYPE_TEXT	20		Used for picking a related entity.
Income Exchange Date	IncomeExchangeDate	DTYPE_DATE	7		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Job Title	JobTitle	DTYPE_TEXT	75		
Last Name	LastName	DTYPE_TEXT	50	Y	
M/F	MF	DTYPE_TEXT	30		Bound by SEX_MF
M/M	MM	DTYPE_TEXT	15		Bound by MR_MS
Manager First Name	ManagerFirstName	DTYPE_TEXT	50		
Manager Id	ManagerId	DTYPE_ID	15		
Manager Integration Id	ManagerIntegrationId	DTYPE_TEXT	30		
Manager Last Name	ManagerLastName	DTYPE_TEXT	50		Used for picking a related entity
Manager Middle Name	ManagerMiddleName	DTYPE_TEXT	50		
Manager Primary Organization	ManagerPrimaryOrganization	DTYPE_TEXT	100		
Manager Primary Organization Id	ManagerPrimaryOrganizationId	DTYPE_ID	15		
Manager UId	ManagerUId	DTYPE_TEXT	100		
Marital Status	MaritalStatus	DTYPE_TEXT	30		Bound by MARITAL_STA TUS

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Middle Name	MiddleName	DTYPE_TEXT	50		
Mother Maiden Name	MotherMaidenName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100		
Person UId	PersonUId	DTYPE_TEXT	100		
Preferred Communications	PreferredCommunications	DTYPE_TEXT	30		Bound by OFFER_MEDIA
Preferred Language Code	PreferredLanguageCode	DTYPE_TEXT	20		
Privacy Code	PrivacyCode	DTYPE_TEXT	30		Bound by PRIVACY_CODE
Social Security Number	SocialSecurityNumber	DTYPE_TEXT	11		
Status	Status	DTYPE_TEXT	30		Bound by CONTACT_STATUS
Suppress All Calls	SuppressAllCalls	DTYPE_TEXT	1		
Suppress All Emails	SuppressAllEmails	DTYPE_TEXT	1		
Suppress All Faxes	SuppressAllFaxes	DTYPE_TEXT	1		
Suppress All Mailings	SuppressAllMailings	DTYPE_TEXT	1		
Time Zone Id	TimeZoneId	DTYPE_ID	15		
Time Zone Name	TimeZoneName	DTYPE_TEXT	100		Used for picking a related entity.
Work Phone #	WorkPhone	DTYPE_TEXT	40		
Work Phone Extension	WorkPhoneExtension	DTYPE_TEXT	10		

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
operation	operation	DTYPE_TEXT	30		System field
Account					
Account	Account	DTYPE_TEXT	100	Y	
Account Location	AccountLocation	DTYPE_TEXT	50		
Account Organization	AccountOrganization	DTYPE_TEXT	100		
Account Party UId	AccountPartyUId	DTYPE_TEXT	100		
Account Row Id	AccountRowId	DTYPE_ID	15		
Account Status	AccountStatus	DTYPE_TEXT	24		Bound by ACCOUNT_STATUS
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Related Account Integration Id	RelatedAccountIntegrationId	DTYPE_TEXT	30		
operation	operation	DTYPE_TEXT	30		
Alternate Phone					
Alternate Phone #	AlternatePhone	DTYPE_TEXT	40	Y	
Alternate Phone Id	AlternatePhoneId	DTYPE_ID	15		
Alternate Phone Integration Id	AlternatePhoneIntegrationId	DTYPE_TEXT	30		
Alternate Phone Medium Type	AlternatePhoneMediumType	DTYPE_TEXT	30		
Alternate Phone Name	AlternatePhoneName	DTYPE_TEXT	50		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
operation	operation	DTYPE_TEXT	30		System field

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Communication Address					
Alternate Email Address	AlternateEmailAddress	DTYPE_TEXT	100	Y	
Communication Address Id	CommunicationAddressId	DTYPE_ID	15		
Communication Address Integration Id	CommunicationAddressIntegrationId	DTYPE_TEXT	30		
Communication Address Medium Type	CommunicationAddressMediumType	DTYPE_TEXT	30		
Communication Address Name	CommunicationAddressName	DTYPE_TEXT	50	Y	
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
operation	operation	DTYPE_TEXT	30		System field
Household					
Household Id	HouseholdId	DTYPE_ID	15		
Household Integration Id	HouseholdIntegrationId	DTYPE_TEXT	30		
Household Name	HouseholdName	DTYPE_TEXT	50		
Household Number	HouseholdNumber	DTYPE_TEXT	100		
Household Party UId	HouseholdPartyUId	DTYPE_TEXT	100		
Household Status	HouseholdStatus	DTYPE_TEXT	30		Bound by HOUSEHOLD_ STATUS
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
operation	operation	DTYPE_TEXT	30		

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
INS Personal Address					
INS Personal Address Id	INSPersonalAddressId	DTYPE_ID	15		
INS Personal Address Name	INSPersonalAddressName	DTYPE_TEXT	100		
INS Personal Address Type	INSPersonalAddressType	DTYPE_TEXT	30		
INS Personal City	INSPersonalCity	DTYPE_TEXT	50	Y	
INS Personal Country	INSPersonalCountry	DTYPE_TEXT	30		Bound by COUNTRY
INS Personal County	INSPersonalCounty	DTYPE_TEXT	50		
INS Personal Integration Id	INSPersonalIntegrationId	DTYPE_TEXT	30		
INS Personal Postal Code	INSPersonalPostalCode	DTYPE_TEXT	30		
INS Personal State	INSPersonalState	DTYPE_TEXT	10		Bound by STATE_ABBRE V
INS Personal Street Address	INSPersonalStreetAddress	DTYPE_TEXT	200	Y	
INS Personal Street Address 2	INSPersonalStreetAddress2	DTYPE_TEXT	100		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
operation	operation	DTYPE_TEXT	30		System field
Organization					
Division	Division	DTYPE_TEXT	255		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Position	Position	DTYPE_TEXT	50		
Position Id	PositionId	DTYPE_ID	15		
Position Integration Id	PositionIntegrationId	DTYPE_TEXT	30		
Sales Rep	SalesRep	DTYPE_TEXT	50		
operation	operation	DTYPE_TEXT	30		System field
Position					
Division	Division	DTYPE_TEXT	255		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Position	Position	DTYPE_TEXT	50		
Position Id	PositionId	DTYPE_ID	15		
Position Integration Id	PositionIntegrationId	DTYPE_TEXT	30		
Sales Rep	SalesRep	DTYPE_TEXT	50		
operation	operation	DTYPE_TEXT	30		System field
Action					
Account Id	AccountId	DTYPE_ID	15		
Account Location	AccountLocation	DTYPE_TEXT	50		
Account Name	AccountName	DTYPE_TEXT	100		Used for picking a related entity.
Audience	Audience	DTYPE_TEXT	30		Bound by ACTIVITY_AUDIENCE_TYPE
Category	Category	DTYPE_TEXT	30		Bound by TODO_TYPE

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Description	Description	DTYPE_TEXT	100		
Done	Done	DTYPE_UTCDATETIME	7		
Done Flag	DoneFlag	DTYPE_TEXT	1		
Due	Due	DTYPE_UTCDATETIME	7		
Duration Minutes	DurationMinutes	DTYPE_INTEGER	22		Bound by APPOINTMENT_DURATION
Id	Id	DTYPE_ID	15		
Integration Id	IntegrationId	DTYPE_TEXT	30		
MeetingLocation	MeetingLocation	DTYPE_TEXT	100		
No Sooner Than Date	NoSoonerThanDate	DTYPE_UTCDATETIME	7		
Planned	Planned	DTYPE_UTCDATETIME	7		
Started	Started	DTYPE_UTCDATETIME	7		
Status	Status	DTYPE_TEXT	30		Bound by EVENT_STATUS
Type	Type	DTYPE_TEXT	30	Y	Bound by TODO_TYPE
operation	operation	DTYPE_TEXT	30		System field
FINCORP Account					
Account Branch ABA	AccountBranchABA	DTYPE_TEXT	50		Used for picking a related entity.

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Account Branch Id	AccountBranchId	DTYPE_ID	15		
Account Holder	AccountHolder	DTYPE_TEXT	100		Used for picking a related entity.
Account Name	AccountName	DTYPE_TEXT	100		
Account Number	AccountNumber3	DTYPE_TEXT	100	Y	
Account Status	AccountStatus2	DTYPE_TEXT	30		Bound by FINS_CRPST_LFLN_IPFS_PLC_MLOV
Account Type	AccountType	DTYPE_TEXT	30		Bound by FINCORP_PROD_ADMIN_CLASS_MLOV
Acct Gen - Branch Id	AcctGen-BranchId	DTYPE_ID	15		
Available Balance	AvailableBalance	DTYPE_CURRENCY	22		
Billing Option	BillingOption	DTYPE_TEXT	30		Bound by FINCORP_ACCOUNT_BILL_OPTION
Branch	Branch	DTYPE_TEXT	50		
Current Balance	CurrentBalance	DTYPE_CURRENCY	22		
Id	Id	DTYPE_ID	15		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Name	Name	DTYPE_TEXT	100	Y	Used for picking a related entity

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Organization Id	OrganizationId2	DTYPE_ID	15		
Product Id	ProductId	DTYPE_ID	15	Y	
Relationship Limit	RelationshipLimit	DTYPE_CURRENCY	22		
Start Date	StartDate	DTYPE_DATETIME	7		
operation	operation	DTYPE_TEXT	30		System field

Household

CIF Household messages are used in insert, query, update, and delete operations to maintain Household information. LookUpHousehold and LookUpHouseholdByName are used for query operations. UpdateHousehold, InsertHoushold, and DeleteHousehold are message pairs for request and response operations. Figure 21 describes the structure of this message.

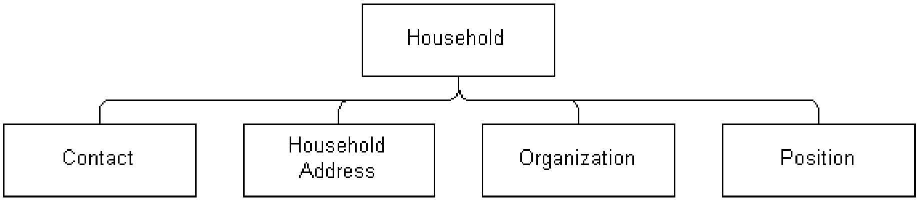


Figure 21. CIF Household Structure

Table 68. Household Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHousehold	Household	LookUpHousehold	1:PartyUId 2: Id
CIFHouseholdRs	Household	LookUpHouseholdResponse	1:PartyUId 2: Id

Table 69. Household Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHouseholdByName	Household	LookUpHouseholdByName	1:PartyUId 2: Id
CIFHouseholdByNameRs	Household	LookUpHouseholdByNameResponse	1:PartyUId 2: Id

Table 70. Household Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHouseholdUpdate	Household	UpdateHousehold	1:PartyUId 2: Id
CIFHouseholdUpdateRs	Household	UpdateHouseholdResponse	1:PartyUId 2: Id

Table 71. Household Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHouseholdInsert	Household	InsertHousehold	1:PartyUId 2: Id
CIFHouseholdInsertRs	Household	InsertHouseholdResponse	1:PartyUId 2: Id

Table 72. Houshold Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHouseholdDelete	Household	DeleteHousehold	1:PartyUId 2: Id
CIFHouseholdDeleteRs	Household	DeleteHouseholdResponse	1:PartyUId 2: Id

Table 73. Household Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Household					
Alias	Alias	DTYPE_TEXT	50		
Category	Category	DTYPE_TEXT	30		Bound by HOUSEHOLD_ CATEGORIES
Currency Code	CurrencyCode	DTYPE_TEXT	20		Used for picking a related entity.
Fax Number	FaxNumber	DTYPE_TEXT	40		
Household Name	HouseholdName	DTYPE_TEXT	50		

Table 73. Household Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Household Number	HouseholdNumber	DTYPE_TEXT	100		
Household Size	HouseholdSize	DTYPE_INTEGER	22		
Household Wealth	HouseholdWealth	DTYPE_CURRENCY	22		
Id	Id	DTYPE_ID	30		This tag is required for Id cross-referencing. Tag stores incoming external Id.
Income	Income	DTYPE_CURRENCY	7		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Party UId	PartyUId	DTYPE_TEXT	100		
Phone Number	PhoneNumber	DTYPE_TEXT	40		
Privacy Code	PrivacyCode	DTYPE_TEXT	30		Bound by PRIVACY_CODE
Revenue	Revenue	DTYPE_CURRENCY	22		
Segment	Segment	DTYPE_TEXT	30		Bound by HOUSEHOLD_SEGMENT
Status	Status	DTYPE_TEXT	30		Bound by HOUSEHOLD_STATUS
Suppress All Calls	SuppressAllCalls	DTYPE_TEXT	1		
Suppress All Mailings	SuppressAllMailings	DTYPE_TEXT	1		

Table 73. Household Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Type	Type	DTYPE_TEXT	30		Bound by HOUSEHOLD_ TYPE
operation	operation	DTYPE_TEXT	30		
Contact					
Contact Integration Id	ContactIntegrationId	DTYPE_TEXT	30		
Contact Party UId	ContactPartyUId	DTYPE_TEXT	100		
DOB	DOB	DTYPE_DATE	7		
First Name	FirstName	DTYPE_TEXT	50	Y	
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Last Name	LastName	DTYPE_TEXT	50	Y	
M/M	MM	DTYPE_TEXT	15		Bound by MR_MS
Middle Name	MiddleName	DTYPE_TEXT	50		
Person UId	PersonUId	DTYPE_TEXT	100		
Related Contact Id	RelatedContactId	DTYPE_ID	15		
Related Date Entered Household	RelatedDateEnteredHousehold	DTYPE_UTCD ATETIME	7		
Related Date Exited Household	RelatedDateExitedHousehold	DTYPE_UTCD ATETIME	7		
Related Primary Organization Id	RelatedPrimaryOrganizationId	DTYPE_ID	15		
Relation to Household	RelationtoHousehold	DTYPE_TEXT	30		Bound by REL_TO_HOU SEHOLD
SSN	SSN	DTYPE_TEXT	11		

Table 73. Household Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
operation	operation	DTYPE_TEXT	30		System field
Household Address					
Address Integration Id	AddressIntegrationId	DTYPE_TEXT	30		
Address Name	AddressName	DTYPE_TEXT	100		
Address Row Id	AddressRowId	DTYPE_ID	15		
City	City	DTYPE_TEXT	50	Y	
Country	Country	DTYPE_TEXT	30		Bound by COUNTRY
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Postal code	Postalcode	DTYPE_TEXT	30		
State	State	DTYPE_TEXT	10		Bound by STATE_ABBRE V
Street Address	StreetAddress	DTYPE_TEXT	200	Y	
Street Address 2	StreetAddress2	DTYPE_TEXT	100		
operation	operation	DTYPE_TEXT	30		System field
Organization					
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Organization	Organization	DTYPE_TEXT	50		
Organization Id	OrganizationId	DTYPE_ID	30		
Organization Integration Id	OrganizationIntegrationId	DTYPE_TEXT	30		
operation	operation	DTYPE_TEXT	30		System field
Position					

Table 73. Household Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Position	Position	DTYPE_TEXT	50		
Position Division	PositionDivision	DTYPE_TEXT	255		
Position Id	PositionId	DTYPE_ID	15		
Position Integration Id	PositionIntegrationId	DTYPE_TEXT	30		
Team	Team	DTYPE_TEXT	50		
operation	operation	DTYPE_TEXT	30		System field

Personal Address

CIF Personal Address messages are used for requesting Personal Address information. They are a message pair for request and response operations. Figure 22 describes the structure of this message.

Figure 22. CIF Personal Address Structure

Table 74. Personal Address Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFPersonalAddress	Personal Address	LookUpPersonalAddress	1: Street Address, City,State, Postal Code
CIFPersonalAddressRs	Household	InsertHouseholdResponse	1: Street Address, City,State, Postal Code

Table 75. Personal Address Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Personal Address					
Address Description	AddressDescription	DTYPE_TEXT	30		Bound by FIN_CON_ADDRESS_DESC
Address Name	AddressName	DTYPE_TEXT	100		
Address Type	AddressType	DTYPE_TEXT	30		Bound by FIN_CON_ADDRESS_TYPE
Apartment Number	ApartmentNumber	DTYPE_TEXT	5		
City	City	DTYPE_TEXT	50	Y	
Country	Country	DTYPE_TEXT	30		Bound by COUNTRY
County	County	DTYPE_TEXT	50		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Postal Code	PostalCode	DTYPE_TEXT	30		
State	State	DTYPE_TEXT	10		Bound by STATE_ABBREV
Street Address	StreetAddress	DTYPE_TEXT	200	Y	
Street Address 2	StreetAddress2	DTYPE_TEXT	100		
Zip Code	ZipCode	DTYPE_TEXT	30		
operation	operation	DTYPE_TEXT	30		System field

FINCORP Account

CIF FINCORP Account messages are used for requesting FINCORP Account information. They are a message pair for request and response operations. Figure 23 describes the structure of this message.

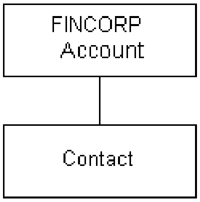


Figure 23. CIF FINCORP Account Structure

Table 76. FINCORP Account Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFFINCORPAccount	FINCORP Account	LookUpFINCORPAccount	1: Account Number
CIFFINCORPAccountRs	FINCORP Account	LookUpFINCORPAccountResponse	1: Account Number

Table 77. FINCORP Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
FINCORP Account					
Account Branch ABA	AccountBranchABA	DTYPE_TEXT	50		Used for picking a related entity
Account Branch Id	AccountBranchId	DTYPE_ID	15		

Table 77. FINCORP Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Account Holder	AccountHolder	DTYPE_TEXT	100		Used for picking a related entity.
Account Name	AccountName	DTYPE_TEXT	100		
Account Number	AccountNumber	DTYPE_TEXT	100	Y	
Account Status	AccountStatus	DTYPE_TEXT	30		Bound by FINS_CRPST_LFLN_IPFS_PLC_MLOV
Account Type	AccountType	DTYPE_TEXT	30		Bound by FINCORP_PROD_ADMIN_CLASS_MLOV
Acct Gen - Branch Id	AcctGen-BranchId	DTYPE_ID	15		
Available Balance	AvailableBalance	DTYPE_CURRENCY	22		
Billing Option	BillingOption	DTYPE_TEXT	30		Bound by FINCORP_ACCOUNT_BILL_OPTION
Branch	Branch	DTYPE_TEXT	50		
CA Next Statement Date	CANextStatementDate	DTYPE_DATE	7		
Commitment Amount	CommitmentAmount	DTYPE_CURRENCY	22		
Current Balance	CurrentBalance	DTYPE_CURRENCY	22		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Name	Name	DTYPE_TEXT	100	Y	Used for picking a related entity.

Table 77. FINCORP Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Organization Id	OrganizationId	DTYPE_ID	15		
Parent ABA Number	ParentABANumber	DTYPE_TEXT	50		
Parent Asset ID	ParentAssetID	DTYPE_ID	15		
Parent Asset Number	ParentAssetNumber	DTYPE_TEXT	100		
Primary Account Id	PrimaryAccountId	DTYPE_ID	15		
Product Id	ProductId	DTYPE_ID	15	Y	
Relationship	Relationship	DTYPE_TEXT	30		
Relationship Limit	RelationshipLimit	DTYPE_CURRENCY	22		
Start Date	StartDate	DTYPE_DATE	7		
operation	operation	DTYPE_TEXT	30		System field
Contact					
Acct Gen - Authorized Users - First Name	AcctGen-AuthorizedUsers-FirstName	DTYPE_TEXT	50	Y	
Acct Gen - Authorized Users - Last Name	AcctGen-AuthorizedUsers-LastName	DTYPE_TEXT	50	Y	
Acct Gen - Authorized Users - Party UId	AcctGen-AuthorizedUsers-PartyUId	DTYPE_TEXT	100		
CL Customer Value	CLCustomerValue	DTYPE_TEXT	30		Bound by CL_ACCTPRFGRP_LOV
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
operation	operation	DTYPE_TEXT	30		System field

Account Profile

CIF Account Profile messages are used in insert, query, update, and delete operations to maintain Account Profile Information. LookUpAccount Profile and LookUptProfileByName are used for query operations. UpdateAccountProfile, InsertAccountProfile, and DeleteAccountProfile are message pairs for request and response operations. Figure 24 describes the structure of this message.

Figure 24. CIF Account Profile Structure

Table 78. Account Profile Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProfile	Account	LookUpAccountProfile	UK1:PartyUId UK2: Id
CIFAccountProfileRs	Account	LookUpAccountProfileResponse	UK1:PartyUId UK2: Id

Table 79. Account Profile Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProfileByName	Account	LookUpAccountProfileByName	UK1:PartyUId UK2: Id
CIFAccountProfileByNameRs	Account	LookUpAccountProfileByNameResponse	UK1:PartyUId UK2: Id

Table 80. Account Profile Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProfileUpdate	Account	UpdateAccountProfile	UK1:PartyUId UK2: Id
CIFAccountProfileUpdateRs	Account	UpdateAccountProfileResponse	UK1:PartyUId UK2: Id

Table 81. Account Profile Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProfileInsert	Account	InsertAccountProfile	UK1:PartyUId UK2: Id
CIFAccountProfileInsertRs	Account	InsertAccountProfileResponse	UK1:PartyUId UK2: Id

Table 82. Account Profile Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProfileInsert	Account	InsertAccountProfile	UK1:PartyUId UK2: Id
CIFAccountProfileInsertRs	Account	InsertAccountProfileResponse	UK1:PartyUId UK2: Id

Table 83. Account Profile Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Account Region	AccountRegion	DTYPE_TEXT	30	N	Use for picking a related entity
Account Status	AccountStatus	DTYPE_TEXT	24	N	Bound by ACCOUNT_STATUS
Account Type Code	AccountTypeCode	DTYPE_TEXT	30	Y	Bound by CUT_ACCOUNT_TYPE
Account Value	AccountValue	DTYPE_TEXT	30	N	Bound by FIN_CON_VALUE_TYPE
Alias	Alias	DTYPE_TEXT	50	N	
Annual Revenue	AnnualRevenue	DTYPE_TEXT	22	N	
Client Flag	ClientFlag	DTYPE_BOOL	1	N	
Competitor	Competitor	DTYPE_BOOL	1	N	
Credit Agency Number	CreditAgencyNumber	DTYPE_TEXT	30	N	
Cumulative Sales	CumulativeSales	DTYPE_NUMBER	22	N	
Currency Code	CurrencyCode	DTYPE_TEXT	15	Y	
Current Volume	CurrentVolume	DTYPE_CURRENCY	22	N	
Dont Call	DontCall	DTYPE_BOOL	1	N	
Drug Spent Currency Code	DrugSpentCurrencyCode	DTYPE_TEXT	30	Y	Use for picking a related entity
Employees	Employees	DTYPE_NUMBER	22	N	

Table 83. Account Profile Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Facility Flag	FacilityFlag	DTYPE_BOOL	1	N	
Federal Tax Number	FederalTaxNumber	DTYPE_TEXT	15	N	
Financing Method	FinancingMethod	DTYPE_TEXT	30	N	Bound by FINCORP_CLIENT_FINANCING
Gross Potential Value	GrossPotentialValue	DTYPE_CURRENCY	22	N	
Group Type Code	GroupTypeCode	DTYPE_TEXT	30	Y	Bound by GROUP_TYPE_CD
Home Page	HomePage	DTYPE_TEXT	100	N	
Integration Id	IntegrationId	DTYPE_TEXT	30	N	
Internal Org Flag	InternalOrgFlag	DTYPE_BOOL	1	Y	
Investor Flag	InvestorFlag	DTYPE_BOOL	1	N	
Last Year Sales	LastYearSales	DTYPE_CURRENCY	22	N	
Legal Status	LegalStatus	DTYPE_TEXT	30	N	Bound by UT_LEGAL_STATUS
Location	Location	DTYPE_TEXT	50	N	
Main Fax Number	MainFaxNumber	DTYPE_PHONE	40	N	
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40	N	
Master Account Id	MasterAccountId	DTYPE_ID	15	N	
Name	Name	DTYPE_TEXT	100	Y	
Partner Flag	PartnerFlag	DTYPE_BOOL	1	N	

Table 83. Account Profile Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Party UId	PartyUId	DTYPE_TEXT	100	Y	
Privacy Code	PrivacyCode	DTYPE_TEXT	30	Y	Bound by PRIVACY_CODE
Reference Flag	ReferenceFlag	DTYPE_BOOL	1	N	
Referenciable	Referenciable	DTYPE_BOOL	1	N	
Revenue Growth	RevenueGrowth	DTYPE_NUMBER	22	N	
Service Provider Flag	ServiceProviderFlag	DTYPE_TEXT	1	N	
Status	Status	DTYPE_TEXT	30	N	Bound by FINS_COMPETITOR_STATUS_MLOV
Total # Meters	TotalMeters	DTYPE_NUMBER	22	N	
Total Potential Volume	TotalPotentialVolume	DTYPE_CURRENCY	22	N	
Type	Type	DTYPE_TEXT	30	Y	Bound by ACCOUNT_TYPE
Unionized	Unionized	DTYPE_BOOL	1	N	
YTD Net Contribution	YTDNetContribution	DTYPE_CURRENCY	22	N	
operation	operation	DTYPE_TEXT	30	N	
Id	Id	DTYPE_ID	15	N	Use for cross-referencing

Account Address

CIF Account Address messages are used in insert, query, update, and delete operations to maintain Account Address Information. LookUpAccount Address and LookUpAccountAddressByName are used for query operations. UpdateAccountAddress, InsertAccountAddress, and DeleteAccountAddress are message pairs for request and response operations. Figure 25 describes the structure of this message.

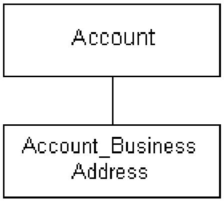


Figure 25. CIF Account Address Structure

Table 84. Account Address Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountAddress	Account	LookUpAccountAddress	UK1:PartyUId UK2: Id
CIFAccountAddressRs	Account	LookUpAccountAddressResponse	UK1:PartyUId UK2: Id

Table 85. Account Address Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountAddressByName	Account	LookUpAccountAddressByName	UK1:PartyUId UK2: Id
CIFAccountAddressByNameRs	Account	LookUpAccountAddressByNameResponse	UK1:PartyUId UK2: Id

Table 86. Account Address Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountAddressUpdate	Account	UpdateAccountAddress	UK1:PartyUId UK2: Id
CIFAccountAddressUpdateRs	Account	UpdateAccountAddressResponse	UK1:PartyUId UK2: Id

Table 87. Account Address Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountAddressInsert	Account	InsertAccountAddress	UK1:PartyUId UK2: Id
CIFAccountAddressInsertRs	Account	InsertAccountAddressResponse	UK1:PartyUId UK2: Id

Table 88. Account Address Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountAddressDelete	Account	DeleteAccountAddress	UK1:PartyUId UK2: Id
CIFAccountAddressDeleteRs	Account	DeleteAccountAddressResponse	UK1:PartyUId UK2: Id

Table 89. Account Address Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Account					
Account Region	AccountRegion	DTYPE_TEXT	30	N	Use for picking a related entity
Account Status	AccountStatus	DTYPE_TEXT	24	N	Bound by ACCOUNT_STATUS
Account Type Code	AccountTypeCode	DTYPE_TEXT	30	Y	Bound by CUT_ACCOUNT_TYPE
Account Value	AccountValue	DTYPE_TEXT	30	N	Bound by FIN_CON_VALUE_TYPE
Alias	Alias	DTYPE_TEXT	50	N	
Annual Revenue	AnnualRevenue	DTYPE_TEXT	22	N	
Client Flag	ClientFlag	DTYPE_BOOL	1	N	
Competitor	Competitor	DTYPE_BOOL	1	N	
Credit Agency Number	CreditAgencyNumber	DTYPE_TEXT	30	N	
Cumulative Sales	CumulativeSales	DTYPE_NUMBER	22	N	
Currency Code	CurrencyCode	DTYPE_TEXT	15	Y	
Current Volume	CurrentVolume	DTYPE_CURRENCY	22	N	
Dont Call	DontCall	DTYPE_BOOL	1	N	
Drug Spent Currency Code	DrugSpentCurrencyCode	DTYPE_TEXT	30	Y	Use for picking a related entity

Table 89. Account Address Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Employees	Employees	DTYPE_NUMBER	22	N	
Facility Flag	FacilityFlag	DTYPE_BOOL	1	N	
Federal Tax Number	FederalTaxNumber	DTYPE_TEXT	15	N	
Financing Method	FinancingMethod	DTYPE_TEXT	30	N	Bound by FINCORP_CLIENT_FINANCING
Gross Potential Value	GrossPotentialValue	DTYPE_CURRENCY	22	N	
Group Type Code	GroupTypeCode	DTYPE_TEXT	30	Y	Bound by GROUP_TYPE_CD
Home Page	HomePage	DTYPE_TEXT	100	N	
Integration Id	IntegrationId	DTYPE_TEXT	30	N	
Internal Org Flag	InternalOrgFlag	DTYPE_BOOL	1	Y	
Investor Flag	InvestorFlag	DTYPE_BOOL	1	N	
Last Year Sales	LastYearSales	DTYPE_CURRENCY	22	N	
Legal Status	LegalStatus	DTYPE_TEXT	30	N	Bound by UT_LEGAL_STATUS
Location	Location	DTYPE_TEXT	50	N	
Main Fax Number	MainFaxNumber	DTYPE_PHONE	40	N	
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40	N	
Master Account Id	MasterAccountId	DTYPE_ID	15	N	

Table 89. Account Address Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Name	Name	DTYPE_TEXT	100	Y	
Partner Flag	PartnerFlag	DTYPE_BOOL	1	N	
Party UId	PartyUId	DTYPE_TEXT	100	Y	
Privacy Code	PrivacyCode	DTYPE_TEXT	30	Y	Bound by PRIVACY_CODE
Reference Flag	ReferenceFlag	DTYPE_BOOL	1	N	
Referenciable	Referenciable	DTYPE_BOOL	1	N	
Revenue Growth	RevenueGrowth	DTYPE_NUMBER	22	N	
Service Provider Flag	ServiceProviderFlag	DTYPE_TEXT	1	N	
Status	Status	DTYPE_TEXT	30	N	Bound by FINS_COMPETITOR_STATUS_MLOV
Total # Meters	TotalMeters	DTYPE_NUMBER	22	N	
Total Potential Volume	TotalPotentialVolume	DTYPE_CURRENCY	22	N	
Type	Type	DTYPE_TEXT	30	Y	Bound by ACCOUNT_TYPE
Unionized	Unionized	DTYPE_BOOL	1	N	
YTD Net Contribution	YTDNetContribution	DTYPE_CURRENCY	22	N	
operation	operation	DTYPE_TEXT	30	N	
Id	Id	DTYPE_ID	15	N	For cross-referencing

Table 89. Account Address Integration Component Fields

Name	XML Tag	Data Type	Length	Requir ed	Notes
Account_Business Address					
Address Id	AddressId	DTYPE_ID	30	Y	
Address Integration Id	AddressIntegrationId	DTYPE_TEXT	30	N	
Bill Address Flag	BillAddressFlag	DTYPE_TEXT	1	N	
City	City	DTYPE_TEXT	50	N	
Country	Country	DTYPE_TEXT	50	N	Use for pick a related entity
Email Address	EmailAddress	DTYPE_TEXT	50	N	
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1	N	
Postal Code	PostalCode	DTYPE_TEXT	30	N	
Ship Address Flag	ShipAddressFlag	DTYPE_TEXT	1	N	
State	State	DTYPE_TEXT	10	N	Bound by STATE_ABBRE V
Street Address	StreetAddress	DTYPE_TEXT	200	N	
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	

Account Activity

CIF Account Activity messages are used in insert, query, update, and delete operations to maintain Account Activity Information. LookUpAccountActivity and LookUpAccountActivityByName are used for query operations. UpdateAccountActivity, InsertAccountActivity, and DeleteAccountActivity are message pairs for request and response operations. Figure 26 describes the structure of this message.

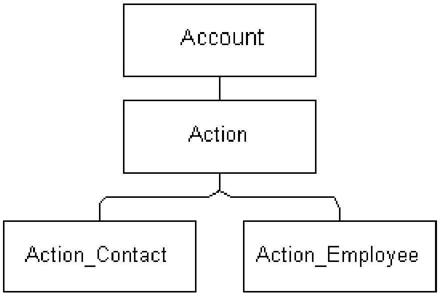


Figure 26. CIF Account Activity Structure

Table 90. Account Activity Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountActivity	Account	LookUpAccountActivity	UK1:PartyUId UK2: Id
CIFAccountActivityRs	Account	LookUpAccountActivityResponse	UK1:PartyUId UK2: Id

Table 91. Account Activity Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountActivityByName	Account	LookUpAccountActivityByNames	UK1:PartyUId UK2: Id
CIFAccountActivityByNameRs	Account	LookUpAccountActivityByNameResponse	UK1:PartyUId UK2: Id

Table 92. Account Activity Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountActivityUpdate	Account	UpdateAccountActivitys	UK1:PartyUId UK2: Id
CIFAccountActivityUpdateRs	Account	UpdateAccountActivityResponse	UK1:PartyUId UK2: Id

Table 93. Account Activity Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountActivityInsert	Account	InsertAccountActivity	UK1:PartyUId UK2: Id
CIFAccountActivityInsertRs	Account	InsertAccountActivityResponse	UK1:PartyUId UK2: Id

Table 94. Account Activity Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountActivityDelete	Account	DeleteAccountActivity	UK1:PartyUId UK2: Id
CIFAccountActivityDeleteRs	Account	DeleteAccountActivityResponse	UK1:PartyUId UK2: Id

Table 95. Account Activity Integration Component Fields

Name	XML Tag	Type	Length	Required	Notes
Account					
Account Region	AccountRegion	DTYPE_TEXT	30	N	Use for picking a related entity
Account Status	AccountStatus	DTYPE_TEXT	24	N	Bound by ACCOUNT_STATUS
Account Type Code	AccountTypeCode	DTYPE_TEXT	30	Y	Bound by CUT_ACCOUNT_TYPE
Account Value	AccountValue	DTYPE_TEXT	30	N	Bound by FIN_CON_VALUE_TYPE
Alias	Alias	DTYPE_TEXT	50	N	
Annual Revenue	AnnualRevenue	DTYPE_TEXT	22	N	
Client Flag	ClientFlag	DTYPE_BOOL	1	N	
Competitor	Competitor	DTYPE_BOOL	1	N	
Credit Agency Number	CreditAgencyNumber	DTYPE_TEXT	30	N	
Cumulative Sales	CumulativeSales	DTYPE_NUMBER	22	N	
Currency Code	CurrencyCode	DTYPE_TEXT	15	Y	
Current Volume	CurrentVolume	DTYPE_CURRENCY	22	N	
Dont Call	DontCall	DTYPE_BOOL	1	N	
Drug Spent Currency Code	DrugSpentCurrencyCode	DTYPE_TEXT	30	Y	Use for picking a related entity

Table 95. Account Activity Integration Component Fields

Name	XML Tag	Type	Length	Required	Notes
Employees	Employees	DTYPE_NUMBER	22	N	
Facility Flag	FacilityFlag	DTYPE_BOOL	1	N	
Federal Tax Number	FederalTaxNumber	DTYPE_TEXT	15	N	
Financing Method	FinancingMethod	DTYPE_TEXT	30	N	Bound by FINCORP_CLIENT_FINANCING
Gross Potential Value	GrossPotentialValue	DTYPE_CURRENCY	22	N	
Group Type Code	GroupTypeCode	DTYPE_TEXT	30	Y	Bound by GROUP_TYPE_CD
Home Page	HomePage	DTYPE_TEXT	100	N	
Integration Id	IntegrationId	DTYPE_TEXT	30	N	
Internal Org Flag	InternalOrgFlag	DTYPE_BOOL	1	Y	
Investor Flag	InvestorFlag	DTYPE_BOOL	1	N	
Last Year Sales	LastYearSales	DTYPE_CURRENCY	22	N	
Legal Status	LegalStatus	DTYPE_TEXT	30	N	Bound by UT_LEGAL_STATUS
Location	Location	DTYPE_TEXT	50	N	
Main Fax Number	MainFaxNumber	DTYPE_PHONE	40	N	
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40	N	
Master Account Id	MasterAccountId	DTYPE_ID	15	N	

Table 95. Account Activity Integration Component Fields

Name	XML Tag	Type	Length	Required	Notes
Name	Name	DTYPE_TEXT	100	Y	
Partner Flag	PartnerFlag	DTYPE_BOOL	1	N	
Party UId	PartyUId	DTYPE_TEXT	100	Y	
Privacy Code	PrivacyCode	DTYPE_TEXT	30	Y	Bound by PRIVACY_CODE
Reference Flag	ReferenceFlag	DTYPE_BOOL	1	N	
Referenciable	Referenciable	DTYPE_BOOL	1	N	
Revenue Growth	RevenueGrowth	DTYPE_NUMBER	22	N	
Service Provider Flag	ServiceProviderFlag	DTYPE_TEXT	1	N	
Status	Status	DTYPE_TEXT	30	N	Bound by FINS_COMPETITOR_STATUS_MLOV
Total # Meters	TotalMeters	DTYPE_NUMBER	22	N	
Total Potential Volume	TotalPotentialVolume	DTYPE_CURRENCY	22	N	
Type	Type	DTYPE_TEXT	30	Y	Bound by ACCOUNT_TYPE
Unionized	Unionized	DTYPE_BOOL	1	N	
YTD Net Contribution	YTDNetContribution	DTYPE_CURRENCY	22	N	
operation	operation	DTYPE_TEXT	30	N	
Id	Id	DTYPE_ID	15	N	For cross-referencing

Table 95. Account Activity Integration Component Fields

Name	XML Tag	Type	Length	Required	Notes
Action					
Alarm	Alarm	DTYPE_BOOL	1	N	
Assignment Excluded	AssignmentExcluded	DTYPE_BOOL	1	N	
Associated Cost	AssociatedCost	DTYPE_CURRENCY	22	N	
Billable Flag	BillableFlag	DTYPE_BOOL	1	N	
Call Id	CallId	DTYPE_TEXT	30	N	
Category	Category	DTYPE_TEXT	30	Y	Use for picking a related entity
Class	Class	DTYPE_TEXT	30	N	Bound by FS_ACTIVITY_CLASS
Comment	Comment	DTYPE_TEXT	“1,500”	N	
Description	Description	DTYPE_TEXT	100	N	
Display	Display	DTYPE_TEXT	30	N	
Done	Done	DTYPE_UTCDATETIME	7	N	
Done Flag	DoneFlag	DTYPE_BOOL	1	N	
Due	Due	DTYPE_UTCDATETIME	7	N	
Duration Minutes	DurationMinutes	DTYPE_INTEGER	22	N	Bound by APPOINTMENT_DURATION
Expense Related Flag	ExpenseRelatedFlag	DTYPE_BOOL	1	N	
Id	Id	DTYPE_TEXT	100	Y	

Table 95. Account Activity Integration Component Fields

Name	XML Tag	Type	Length	Required	Notes
Integration Id	IntegrationId	DTYPE_TEXT	30	N	
No Sooner Than Date	NoSoonerThanDate	DTYPE_UTCDATETIME	7	N	
Objective	Objective	DTYPE_TEXT	30	N	
Owned By	OwnedBy	DTYPE_TEXT	30	Y	Use for pick a related entity
Owned By Id	OwnedById	DTYPE_TEXT	30	Y	
Primary Owned By	PrimaryOwnedBy	DTYPE_TEXT	50	N	Use for pick a related entity
Primary Owner Id	PrimaryOwnerId	DTYPE_ID	30	N	
Priority	Priority	DTYPE_TEXT	30	N	Bound by ACTIVITY_PRIORITY
Private	Private	DTYPE_BOOL	1	N	
Repeating	Repeating	DTYPE_BOOL	1	N	
Repeating Expires	RepeatingExpires	DTYPE_DATETIME	7	N	
Status	Status	DTYPE_TEXT	30	N	Bound by EVENT_STATUS
Type	Type2	DTYPE_TEXT	30	Y	Bound by TODO_TYPE
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	
Action_Contact					
Company Identifier	CompanyIdentifier	DTYPE_TEXT	15	N	

Table 95. Account Activity Integration Component Fields

Name	XML Tag	Type	Length	Requir ed	Notes
Contact First Name	ContactFirstName	DTYPE_TEXT	50	Y	
Contact Last Name	ContactLastName	DTYPE_TEXT	50	Y	
Employee Number	EmployeeNumber	DTYPE_TEXT	30	N	
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1	N	
Personal Contact	PersonalContact	DTYPE_BOOL	1	N	
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	
Action_Employee					
Emp #	Emp	DTYPE_TEXT	30	N	
Employee First Name (MM)	EmployeeFirstNameMM	DTYPE_TEXT	50	Y	
Employee Job Title (MM)	EmployeeJobTitleMM	DTYPE_TEXT	75	N	
Employee Last Name (MM)	EmployeeLastNameMM	DTYPE_TEXT	50	Y	
Employee Login Name (MM)	EmployeeLoginNameMM	DTYPE_TEXT	50	Y	
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1	N	
Private	Private	DTYPE_BOOL	1	Y	
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	

Account Product

CIF Account Products messages are used in insert, query, update, and delete operations to maintain Account Product (Financial Account) Information. LookUpAccountProduct and LookUpAccountProductByName are used for query operations. UpdateAccountProduct, InsertAccountProduct, and DeleteAccountProduct are message pairs for request and response operations. Figure 27 describes the structure of this message.

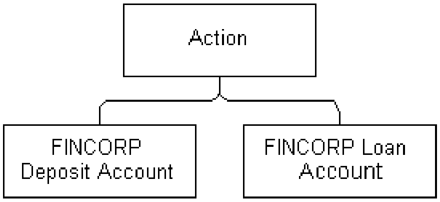


Figure 27. CIF Account Products Structure

Table 96. Account Product Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProduct	Account	LookUpAccountProduct	UK1:PartyUId UK2: Id
CIFAccountProductRs	Account	LookUpAccountProductResponse	UK1:PartyUId UK2: Id

Table 97. Account Product Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProductByName	Account	LookUpAccountProductByName	UK1:PartyUId UK2: Id
CIFAccountProductByNameRs	Account	LookUpAccountProductByNameResponse	UK1:PartyUId UK2: Id

Table 98. Account Product Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProductUpdate	Account	UpdateAccountProduct	UK1:PartyUId UK2: Id
CIFAccountProductUpdateRs	Account	UpdateAccountProductResponse	UK1:PartyUId UK2: Id

Table 99. Account Product Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProductInsert	Account	InsertAccountProduct	UK1:PartyUId UK2: Id
CIFAccountProductInsertRs	Account	InsertAccountProductResponse	UK1:PartyUId UK2: Id

Table 100. Account Product Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProductDelete	Account	DeleteAccountProduct	UK1:PartyUId UK2: Id
CIFAccountProductDeleteRs	Account	DeleteAccountProductResponse	UK1:PartyUId UK2: Id

Table 101. Account Product Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Account					
Account Region	AccountRegion	DTYPE_TEXT	30	N	Use for picking a related entity
Account Status	AccountStatus	DTYPE_TEXT	24	N	Bound by ACCOUNT_STATUS
Account Type Code	AccountTypeCode	DTYPE_TEXT	30	Y	Bound by CUT_ACCOUNT_TYPE
Account Value	AccountValue	DTYPE_TEXT	30	N	Bound by FIN_CON_VALUE_TYPE
Alias	Alias	DTYPE_TEXT	50	N	
Annual Revenue	AnnualRevenue	DTYPE_TEXT	22	N	
Client Flag	ClientFlag	DTYPE_BOOL	1	N	
Competitor	Competitor	DTYPE_BOOL	1	N	
Credit Agency Number	CreditAgencyNumber	DTYPE_TEXT	30	N	
Cumulative Sales	CumulativeSales	DTYPE_NUMBER	22	N	
Currency Code	CurrencyCode	DTYPE_TEXT	15	Y	
Current Volume	CurrentVolume	DTYPE_CURRENCY	22	N	
Dont Call	DontCall	DTYPE_BOOL	1	N	
Drug Spent Currency Code	DrugSpentCurrencyCode	DTYPE_TEXT	30	Y	Use for picking a related entity

Table 101. Account Product Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Employees	Employees	DTYPE_NUMBER	22	N	
Facility Flag	FacilityFlag	DTYPE_BOOL	1	N	
Federal Tax Number	FederalTaxNumber	DTYPE_TEXT	15	N	
Financing Method	FinancingMethod	DTYPE_TEXT	30	N	Bound by FINCORP_CLIENT_FINANCING
Gross Potential Value	GrossPotentialValue	DTYPE_CURRENCY	22	N	
Group Type Code	GroupTypeCode	DTYPE_TEXT	30	Y	Bound by GROUP_TYPE_CD
Home Page	HomePage	DTYPE_TEXT	100	N	
Integration Id	IntegrationId	DTYPE_TEXT	30	N	
Internal Org Flag	InternalOrgFlag	DTYPE_BOOL	1	Y	
Investor Flag	InvestorFlag	DTYPE_BOOL	1	N	
Last Year Sales	LastYearSales	DTYPE_CURRENCY	22	N	
Legal Status	LegalStatus	DTYPE_TEXT	30	N	Bound by UT_LEGAL_STATUS
Location	Location	DTYPE_TEXT	50	N	
Main Fax Number	MainFaxNumber	DTYPE_PHONE	40	N	
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40	N	
Master Account Id	MasterAccountId	DTYPE_ID	15	N	

Table 101. Account Product Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Name	Name	DTYPE_TEXT	100	N	
Partner Flag	PartnerFlag	DTYPE_BOOL	1	N	
Party UId	PartyUId	DTYPE_TEXT	100	Y	
Privacy Code	PrivacyCode	DTYPE_TEXT	30	Y	Bound by PRIVACY_CODE
Reference Flag	ReferenceFlag	DTYPE_BOOL	1	N	
Referenciable	Referenciable	DTYPE_BOOL	1	N	
Revenue Growth	RevenueGrowth	DTYPE_NUMBER	22	N	
Service Provider Flag	ServiceProviderFlag	DTYPE_TEXT	1	N	
Status	Status	DTYPE_TEXT	30	N	Bound by FINS_COMPETITOR_STATUS_MLOV
Total # Meters	TotalMeters	DTYPE_NUMBER	22	N	
Total Potential Volume	TotalPotentialVolume	DTYPE_CURRENCY	22	N	
Type	Type	DTYPE_TEXT	30	N	Bound by ACCOUNT_TYPE
Unionized	Unionized	DTYPE_BOOL	1	N	
YTD Net Contribution	YTDNetContribution	DTYPE_CURRENCY	22	N	
operation	operation	DTYPE_TEXT	30	N	
Id	Id	DTYPE_ID	15	N	For cross-referencing

Table 101. Account Product Integration Component Fields

Name	XML Tag	Data Type	Length	Requir ed	Notes
FINCORP Deposit Account					
Account Branch Number	AccountBranchNumber	DTYPE_TEXT	100	N	
Account Category	AccountCategory	DTYPE_TEXT	30	N	Bound by FINS_ASSET_P ROD_CLASS_ MLOV
Account Number	AccountNumber	DTYPE_TEXT	100	Y	
Account Status	AccountStatus	DTYPE_TEXT	30	N	Bound by FINS_CRPST_L FLN_IPFS_PLC _MLOV
Branch	Branch	DTYPE_TEXT	50	N	
Current Balance	CurrentBalance	DTYPE_CURR ENCY	22	N	
Id	Id	DTYPE_TEXT	100	N	
Integration Id	Integration Id	DTYPE_TEXT	30	N	
Name	Name	DTYPE_TEXT	100	Y	Use for pick a related entity
Product Id	ProductId	DTYPE_ID	100	Y	
Type	Type	DTYPE_TEXT	30	N	Bound by FINS_OWNER SHIP_TYPE_M LOV
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	
FINCORP Loan Account					
Account Branch Number	AccountBranchNumber	DTYPE_TEXT	30	N	

Table 101. Account Product Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Account Category	AccountCategory	DTYPE_TEXT	30	N	Bound by FINS_ASSET_PROD_CLASS_MLOV
Account Number	AccountNumber	DTYPE_TEXT	100	Y	
Account Status	AccountStatus	DTYPE_TEXT	30	N	Bound by FINS_CRPST_LFLN_IPFS_PLC_MLOV
Current Balance	CurrentBalance	DTYPE_CURRENCY	22	N	
Id	Id	DTYPE_TEXT	30	N	
Integration Id	IntegrationId	DTYPE_TEXT	30	N	
Name	Name	DTYPE_TEXT	100	Y	Use for pick a related entity
Product Id	ProductId	DTYPE_ID	100	Y	Use for pick a related entity
Type	Type	DTYPE_TEXT	30	N	FINS_OWNERSHIP_TYPE_MLOV
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	

Account Relationship

CIF Account Relationship messages are used in insert, query, update, and delete operations to maintain Account Relationships. LookUpAccountRelationship and LookUpAccountRelationshipByName are used for query operations. UpdateAccountRelationship, InsertAccountRelationship, and DeleteAccountRelationship are message pairs for request and response operations. Figure 28 describes the structure of this message.

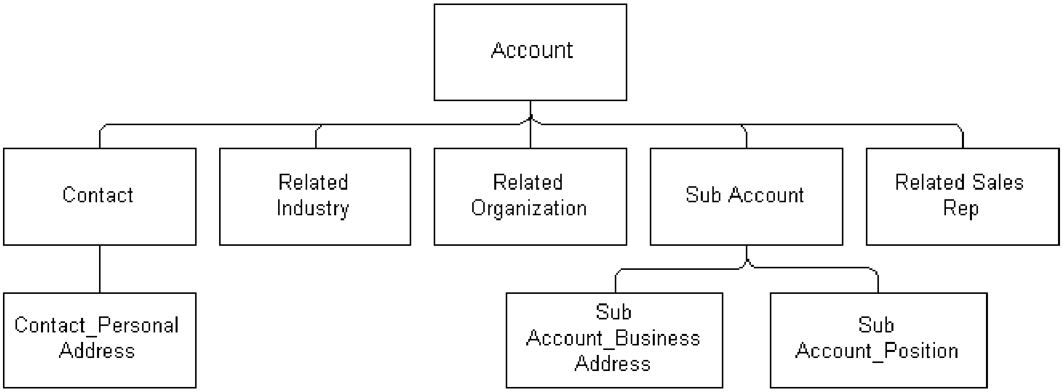


Figure 28. CIF Account Relationship Structure

Table 102. Account Relationship Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountRelationship	Account	LookUpAccountRelationship	UK1:PartyUId UK2: Id
CIFAccountRelationshipRs	Account	LookUpAccountRelationshipResponse	UK1:PartyUId UK2: Id

Table 103. Account Relationship Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountRelationshipByName	Account	LookUpAccountRelationshipByName	UK1:PartyUId UK2: Id
CIFAccountRelationshipByNameRs	Account	LookUpAccountRelationshipByNameResponse	UK1:PartyUId UK2: Id

Table 104. Account Relationship Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountRelationshipUpdate	Account	UpdateAccountRelationship	UK1:PartyUId UK2: Id
CIFAccountRelationshipUpdateRs	Account	UpdateAccountRelationshipResponse	UK1:PartyUId UK2: Id

Table 105. Account Relationship Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountRelationshipInsert	Account	InsertAccountRelationship	UK1:PartyUId UK2: Id
CIFAccountRelationshipInsertRs	Account	InsertAccountRelationshipResponse	UK1:PartyUId UK2: Id

Table 106. Account Relationship Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountRelationshipDelete	Account	DeleteAccountRelationship	UK1:PartyUId UK2: Id
CIFAccountRelationshipDeleteRs	Account	DeleteAccountRelationshipResponse	UK1:PartyUId UK2: Id

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Account					
Account Region	AccountRegion	DTYPE_TEXT	30	N	Use for picking a related entity
Account Status	AccountStatus	DTYPE_TEXT	24	N	Bound by ACCOUNT_STATUS
Account Type Code	AccountTypeCode	DTYPE_TEXT	30	Y	Bound by CUT_ACCOUNT_TYPE
Account Value	AccountValue	DTYPE_TEXT	30	N	Bound by FIN_CON_VALUE_TYPE
Alias	Alias	DTYPE_TEXT	50	N	
Annual Revenue	AnnualRevenue	DTYPE_TEXT	22	N	
Client Flag	ClientFlag	DTYPE_BOOL	1	N	
Competitor	Competitor	DTYPE_BOOL	1	N	
Credit Agency Number	CreditAgencyNumber	DTYPE_TEXT	30	N	
Cumulative Sales	CumulativeSales	DTYPE_NUMBER	22	N	
Currency Code	CurrencyCode	DTYPE_TEXT	15	Y	
Current Volume	CurrentVolume	DTYPE_CURRENCY	22	N	
Dont Call	DontCall	DTYPE_BOOL	1	N	
Drug Spent Currency Code	DrugSpentCurrencyCode	DTYPE_TEXT	30	Y	Use for picking a related entity

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Employees	Employees	DTYPE_NUMBER	22	N	
Facility Flag	FacilityFlag	DTYPE_BOOL	1	N	
Federal Tax Number	FederalTaxNumber	DTYPE_TEXT	15	N	
Financing Method	FinancingMethod	DTYPE_TEXT	30	N	Bound by FINCORP_CLIENT_FINANCING
Gross Potential Value	GrossPotentialValue	DTYPE_CURRENCY	22	N	
Group Type Code	GroupTypeCode	DTYPE_TEXT	30	Y	Bound by GROUP_TYPE_CD
Home Page	HomePage	DTYPE_TEXT	100	N	
Integration Id	IntegrationId	DTYPE_TEXT	30	N	
Internal Org Flag	InternalOrgFlag	DTYPE_BOOL	1	Y	
Investor Flag	InvestorFlag	DTYPE_BOOL	1	N	
Last Year Sales	LastYearSales	DTYPE_CURRENCY	22	N	
Legal Status	LegalStatus	DTYPE_TEXT	30	N	Bound by UT_LEGAL_STATUS
Location	Location	DTYPE_TEXT	50	N	
Main Fax Number	MainFaxNumber	DTYPE_PHONE	40	N	
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40	N	
Master Account Id	MasterAccountId	DTYPE_ID	15	N	

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Name	Name	DTYPE_TEXT	100	Y	
Partner Flag	PartnerFlag	DTYPE_BOOL	1	N	
Party UId	PartyUId	DTYPE_TEXT	100	Y	
Privacy Code	PrivacyCode	DTYPE_TEXT	30	Y	Bound by PRIVACY_CODE
Reference Flag	ReferenceFlag	DTYPE_BOOL	1	N	
Referenciable	Referenciable	DTYPE_BOOL	1	N	
Revenue Growth	RevenueGrowth	DTYPE_NUMBER	22	N	
Service Provider Flag	ServiceProviderFlag	DTYPE_TEXT	1	N	
Status	Status	DTYPE_TEXT	30	N	Bound by FINS_COMPETITOR_STATUS_MLOV
Total # Meters	TotalMeters	DTYPE_NUMBER	22	N	
Total Potential Volume	TotalPotentialVolume	DTYPE_CURRENCY	22	N	
Type	Type	DTYPE_TEXT	30	N	Bound by ACCOUNT_TYPE
Unionized	Unionized	DTYPE_BOOL	1	N	
YTD Net Contribution	YTDNetContribution	DTYPE_CURRENCY	22	N	
operation	operation	DTYPE_TEXT	30	N	
Id	Id	DTYPE_ID	15	N	For cross-referencing

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Requir ed	Notes
Contact					
Account Id	AccountId	DTYPE_ID	30	N	
Account Integration Id	AccountIntegrationId	DTYPE_TEXT	30	N	
Assistant	Assistant	DTYPE_TEXT	50	N	
Assistant Phone #	AssistantPhone	DTYPE_PHON E	40	N	
Cellular Phone #	CellularPhone	DTYPE_PHON E	40	N	
Fax Phone #	FaxPhone	DTYPE_PHON E	40	N	
First Name	FirstName	DTYPE_TEXT	50	Y	
Home Phone #	HomePhone	DTYPE_TEXT	40	N	
Job Title	JobTitle	DTYPE_TEXT	75	N	
Last Name	LastName	DTYPE_TEXT	50	Y	
M/M	MM	DTYPE_TEXT	15	N	
Middle Name	MiddleName	DTYPE_TEXT	50	N	
Party UId	PartyUId	DTYPE_TEXT	100	Y	
Person UId	PersonUId	DTYPE_ID	30	N	
Preferred Communication Method	PreferredCommunicationMethod	DTYPE_TEXT	20	N	Bound by COMM_METH OD
Privacy Code	PrivacyCode	DTYPE_TEXT	30	Y	Bound by PRIVACY_COD E
Row Id	RowId	DTYPE_ID	100	N	

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Work Phone #	WorkPhone	DTYPE_PHONE	45	N	
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	
Contact_Personal Address					
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1	N	
Personal Address Id	PersonalAddressId	DTYPE_ID	100	Y	
Personal Address Integration Id	PersonalAddressIntegrationId	DTYPE_TEXT	30	N	
Personal City	PersonalCity	DTYPE_TEXT	50	Y	
Personal Country	PersonalCountry	DTYPE_TEXT	50	N	Bound by COUNTRY
Personal Postal Code	PersonalPostalCode	DTYPE_TEXT	30	N	
Personal State	PersonalState	DTYPE_TEXT	10	N	Bound by STATE_ABBREV
Personal Street Address	PersonalStreetAddress	DTYPE_TEXT	200	Y	
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	
Related Industry					
Industry	Industry	DTYPE_TEXT	50	Y	
Industry Id	IndustryId	DTYPE_ID	30	Y	
Industry Integration Id	IndustryIntegrationId	DTYPE_TEXT	30	N	

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1	N	
SIC Code	SICCode	DTYPE_TEXT	30	Y	
Related Organization					
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1	N	
Organization	Organization	DTYPE_TEXT	100	Y	
Organization Id	OrganizationId	DTYPE_ID	100	Y	
Organization Integration Id	OrganizationIntegrationId	DTYPE_TEXT	30	N	
Related Sales Reps					
Division	Division	DTYPE_TEXT	255	Y	Use to pick a related entity
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1	N	
Login	Login	DTYPE_ID	15	N	
Position	Position	DTYPE_TEXT	50	N	
Position Id	PositionId	DTYPE_ID	30	N	
Position Integration Id	PositionIntegrationId	DTYPE_TEXT	30	N	
Sub Account					
Account Status	AccountStatus2	DTYPE_TEXT	30	N	Bound by ACCOUNT_STATUS
Id	Id	DTYPE_ID	100	Y	
Location	Location	DTYPE_TEXT	50	N	
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40	N	

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Requir ed	Notes
Name	Name	DTYPE_TEXT	100	Y	
Party UId	PartyUId	DTYPE_TEXT	30	Y	
Type	Type	DTYPE_TEXT	255	Y	Bound by ACCOUNT_TYPE
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	
Party UId	PartyUId	DTYPE_TEXT	30	N	
Type	Type	DTYPE_TEXT	255	N	
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	
Sub Account_Business Address					
Address Id	AddressId	DTYPE_ID	30	Y	
Address Name	AddressName	DTYPE_TEXT	100	N	
Integration Id	IntegrationId	DTYPE_TEXT	30	N	
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1	N	
Street Address	StreetAddress	DTYPE_TEXT	200	Y	
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	
Sub Account_Position					
Division	Division	DTYPE_TEXT	255	Y	Use to pick the related entity
Division Integration Id	DivisionIntegrationId	DTYPE_TEXT	30	N	
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1	N	

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Position	Position	DTYPE_TEXT	50	N	Bound by POSITION_TYPE
Position Id	PositionId	DTYPE_ID	30	N	
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	

Account

CIF Account messages are used in insert, query, update, and delete operations to maintain Account Profile Information. LookUpAccount Profile and LookUptProfileByName are used for query operations. UpdateAccountProfile, InsertAccountProfile, and DeleteAccountProfile are message pairs for request and response operations. Figure 29 describes the structure of this message.

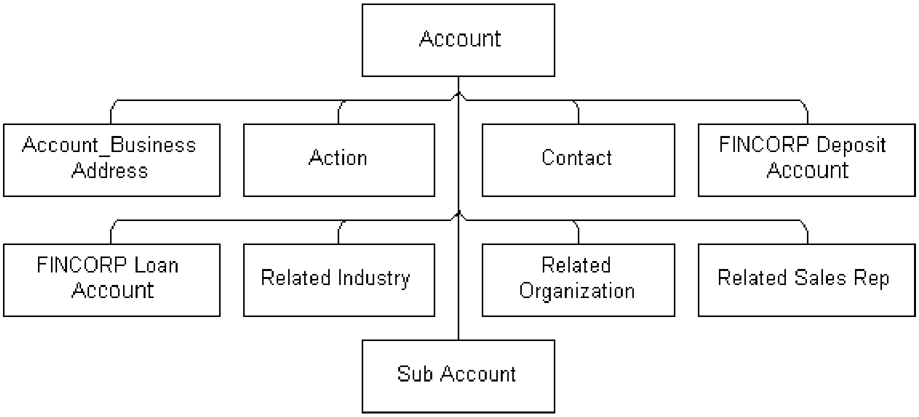


Figure 29. CIF Account Structure

Table 108. Account Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccount	Account	LookUpAccount	UK1:PartyUId UK2: Id
CIFAccountRs	Account	LookUpAccountResponse	UK1:PartyUId UK2: Id

Table 109. Account Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountUpdate	Account	UpdateAccount	UK1:PartyUId UK2: Id
CIFAccountUpdateRs	Account	UpdateAccountResponse	UK1:PartyUId UK2: Id

Table 110. Account Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProductDelete	Account	DeleteAccountProduct	UK1:PartyUId UK2: Id
CIFAccountProductDeleteRs	Account	DeleteAccountProductResponse	UK1:PartyUId UK2: Id

Table 111. Account Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountInsert	Account	InsertAccount	UK1:PartyUId UK2: Id
CIFAccountInsertRs	Account	InsertAccountResponse	UK1:PartyUId UK2: Id

Table 112. Account Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountDelete	Account	DeleteAccount	UK1:PartyUId UK2: Id
CIFAccountDeleteRs	Account	DeleteAccountResponse	UK1:PartyUId UK2: Id

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Requir ed	Notes
Account					
Account Region	AccountRegion	DTYPE_TEXT	30	N	Use for picking a related entity
Account Status	AccountStatus	DTYPE_TEXT	24	N	Bound by ACCOUNT_ST ATUS
Account Type Code	AccountTypeCode	DTYPE_TEXT	30	Y	Bound by CUT_ACCOUN T_TYPE

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Requir ed	Notes
Account Value	AccountValue	DTYPE_TEXT	30	N	Bound by FIN_CON_VALUE_TYPE
Alias	Alias	DTYPE_TEXT	50	N	
Annual Revenue	AnnualRevenue	DTYPE_TEXT	22	N	
Client Flag	ClientFlag	DTYPE_BOOL	1	N	
Competitor	Competitor	DTYPE_BOOL	1	N	
Credit Agency Number	CreditAgencyNumber	DTYPE_TEXT	30	N	
Cumulative Sales	CumulativeSales	DTYPE_NUMBER	22	N	
Currency Code	CurrencyCode	DTYPE_TEXT	15	Y	
Current Volume	CurrentVolume	DTYPE_CURRENCY	22	N	
Dont Call	DontCall	DTYPE_BOOL	1	N	
Drug Spent Currency Code	DrugSpentCurrencyCode	DTYPE_TEXT	30	Y	Use for picking a related entity
Employees	Employees	DTYPE_NUMBER	22	N	
Facility Flag	FacilityFlag	DTYPE_BOOL	1	N	
Federal Tax Number	FederalTaxNumber	DTYPE_TEXT	15	N	
Financing Method	FinancingMethod	DTYPE_TEXT	30	N	Bound by FINCORP_CLIENT_FINANCING
Gross Potential Value	GrossPotentialValue	DTYPE_CURRENCY	22	N	

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Group Type Code	GroupTypeCode	DTYPE_TEXT	30	Y	Bound by GROUP_TYPE_CD
Home Page	HomePage	DTYPE_TEXT	100	N	
Integration Id	IntegrationId	DTYPE_TEXT	30	N	
Internal Org Flag	InternalOrgFlag	DTYPE_BOOL	1	Y	
Investor Flag	InvestorFlag	DTYPE_BOOL	1	N	
Last Year Sales	LastYearSales	DTYPE_CURRENCY	22	N	
Legal Status	LegalStatus	DTYPE_TEXT	30	N	Bound by UT_LEGAL_STATUS
Location	Location	DTYPE_TEXT	50	N	
Main Fax Number	MainFaxNumber	DTYPE_PHONE	40	N	
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40	N	
Master Account Id	MasterAccountId	DTYPE_ID	15	N	
Name	Name	DTYPE_TEXT	100	Y	
Partner Flag	PartnerFlag	DTYPE_BOOL	1	N	
Party UId	PartyUId	DTYPE_TEXT	100	Y	
Privacy Code	PrivacyCode	DTYPE_TEXT	30	Y	Bound by PRIVACY_CODE
Reference Flag	ReferenceFlag	DTYPE_BOOL	1	N	
Referenciable	Referenciable	DTYPE_BOOL	1	N	
Revenue Growth	RevenueGrowth	DTYPE_NUMBER	22	N	

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Service Provider Flag	ServiceProviderFlag	DTYPE_TEXT	1	N	
Status	Status	DTYPE_TEXT	30	N	Bound by FINS_COMPETITOR_STATUS_MLOV
Total # Meters	TotalMeters	DTYPE_NUMBER	22	N	
Total Potential Volume	TotalPotentialVolume	DTYPE_CURRENCY	22	N	
Type	Type	DTYPE_TEXT	30	Y	Bound by ACCOUNT_TYPE
Unionized	Unionized	DTYPE_BOOL	1	N	
YTD Net Contribution	YTDNetContribution	DTYPE_CURRENCY	22	N	
operation	operation	DTYPE_TEXT	30	N	
Id	Id	DTYPE_ID	150	N	For cross-referencing
Account_Business Address					
Address Id	AddressId	DTYPE_ID	30	Y	
Address Integration Id	AddressIntegrationId	DTYPE_TEXT	30	N	
Bill Address Flag	BillAddressFlag	DTYPE_TEXT	1	N	
City	City	DTYPE_TEXT	50	N	
Country	Country	DTYPE_TEXT	50	N	Use for pick a related entity
Email Address	EmailAddress	DTYPE_TEXT	50	N	

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Requir ed	Notes
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1	N	
Postal Code	PostalCode	DTYPE_TEXT	30	N	
Ship Address Flag	ShipAddressFlag	DTYPE_TEXT	1	N	
State	State	DTYPE_TEXT	10	N	Bound by STATE_ABBRE V
Street Address	StreetAddress	DTYPE_TEXT	200	N	
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	
Action					
Alarm	Alarm	DTYPE_BOOL	1	N	
Assignment Excluded	AssignmentExcluded	DTYPE_BOOL	1	N	
Associated Cost	AssociatedCost	DTYPE_CURR ENCY	22	N	
Billable Flag	BillableFlag	DTYPE_BOOL	1	N	
Call Id	CallId	DTYPE_TEXT	30	N	
Category	Category	DTYPE_TEXT	30	Y	Use for picking a related entity
Class	Class	DTYPE_TEXT	30	N	Bound by FS_ACTIVITY_ CLASS
Comment	Comment	DTYPE_TEXT	“1,500”	N	
Description	Description	DTYPE_TEXT	100	N	
Display	Display	DTYPE_TEXT	30	N	
Done	Done	DTYPE_UTCD ATETIME	7	N	

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Done Flag	DoneFlag	DTYPE_BOOL	1	N	
Due	Due	DTYPE_UTCDATETIME	7	N	
Duration Minutes	DurationMinutes	DTYPE_INTEGER	22	N	Bound by APPOINTMENT_DURATION
Expense Related Flag	ExpenseRelatedFlag	DTYPE_BOOL	1	N	
Id	Id	DTYPE_TEXT	100	Y	
Integration Id	IntegrationId	DTYPE_TEXT	30	N	
No Sooner Than Date	NoSoonerThanDate	DTYPE_UTCDATETIME	7	N	
Objective	Objective	DTYPE_TEXT	30	N	
Owned By	OwnedBy	DTYPE_TEXT	30	Y	Use for pick a related entity
Owned By Id	OwnedById	DTYPE_TEXT	30	Y	
Primary Owned By	PrimaryOwnedBy	DTYPE_TEXT	50	N	Use for pick a related entity
Primary Owner Id	PrimaryOwnerId	DTYPE_ID	30	N	
Priority	Priority	DTYPE_TEXT	30	N	Bound by ACTIVITY_PRIORITY
Private	Private	DTYPE_BOOL	1	N	
Repeating	Repeating	DTYPE_BOOL	1	N	
Repeating Expires	RepeatingExpires	DTYPE_DATETIME	7	N	
Status	Status	DTYPE_TEXT	30	N	Bound by EVENT_STATUS

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Type	Type2	DTYPE_TEXT	30	Y	Bound by TODO_TYPE
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	
Contact					
Account Id	AccountId	DTYPE_ID	30	N	
Account Integration Id	AccountIntegrationId	DTYPE_TEXT	30	N	
Assistant	Assistant	DTYPE_TEXT	50	N	
Assistant Phone #	AssistantPhone	DTYPE_PHONE	40	N	
Cellular Phone #	CellularPhone	DTYPE_PHONE	40	N	
Fax Phone #	FaxPhone	DTYPE_PHONE	40	N	
First Name	FirstName	DTYPE_TEXT	50	Y	
Home Phone #	HomePhone	DTYPE_TEXT	40	N	
Job Title	JobTitle	DTYPE_TEXT	75	N	
Last Name	LastName	DTYPE_TEXT	50	Y	
M/M	MM	DTYPE_TEXT	15	N	
Middle Name	MiddleName	DTYPE_TEXT	50	N	
Party UId	PartyUId	DTYPE_TEXT	100	Y	
Person UId	PersonUId	DTYPE_ID	30	Y	
Preferred Communication Method	PreferredCommunicationMethod	DTYPE_TEXT	20	N	Bound by COMM_METHOD

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Requir ed	Notes
Privacy Code	PrivacyCode	DTYPE_TEXT	30	Y	Bound by PRIVACY_COD E
Row Id	RowId	DTYPE_ID	100	N	
Work Phone #	WorkPhone	DTYPE_PHON E	45	N	
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	
FINCORP Deposit Account					
Account Branch Number	AccountBranchNumber	DTYPE_TEXT	100	N	
Account Category	AccountCategory	DTYPE_TEXT	30	N	Bound by FINS_ASSET_P ROD_CLASS_ MLOV
Account Number	AccountNumber	DTYPE_TEXT	100	Y	
Account Status	AccountStatus	DTYPE_TEXT	30	N	Bound by FINS_CRPST_L FLN_IPFS_PLC _MLOV
Branch	Branch	DTYPE_TEXT	50	N	
Current Balance	CurrentBalance	DTYPE_CURR ENCY	22	N	
Id	Id	DTYPE_TEXT	100	N	
Integration Id	Integration Id	DTYPE_TEXT	30	N	
Name	Name	DTYPE_TEXT	100	Y	Use for pick a related entity
Product Id	ProductId	DTYPE_ID	100	Y	

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Notes
Type	Type	DTYPE_TEXT	30	N	Bound by FINS_OWNERSHIP_TYPE_MLOV
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	
FINCORP Loan Account					
Account Branch Number	AccountBranchNumber	DTYPE_TEXT	30	N	
Account Category	AccountCategory	DTYPE_TEXT	30	N	Bound by FINS_ASSET_PROD_CLASS_MLOV
Account Number	AccountNumber	DTYPE_TEXT	100	Y	
Account Status	AccountStatus	DTYPE_TEXT	30	N	Bound by FINS_CRPST_LFLN_IPFS_PLC_MLOV
Current Balance	CurrentBalance	DTYPE_CURRENCY	22	N	
Id	Id	DTYPE_TEXT	30	N	
Integration Id	IntegrationId	DTYPE_TEXT	30	N	
Name	Name	DTYPE_TEXT	100	Y	Use for pick a related entity
Product Id	ProductId	DTYPE_ID	100	Y	Use for pick a related entity
Type	Type	DTYPE_TEXT	30	N	FINS_OWNERSHIP_TYPE_MLOV
operation	operation	DTYPE_TEXT	30	N	

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Requir ed	Notes
searchspec	searchspec	DTYPE_TEXT	250	N	
Related Industry					
Industry	Industry	DTYPE_TEXT	50	Y	
Industry Id	IndustryId	DTYPE_ID	30	Y	
Industry Integration Id	IndustryIntegrationId	DTYPE_TEXT	30	N	
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1	N	
SIC Code	SICCode	DTYPE_TEXT	30	Y	
Related Organization					
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1	N	
Organization	Organization	DTYPE_TEXT	100	Y	
Organization Id	OrganizationId	DTYPE_ID	100	Y	
Organization Integration Id	OrganizationIntegrationId	DTYPE_TEXT	30	N	
Related Sales Reps					
Division	Division	DTYPE_TEXT	255	Y	Use to pick a related entity
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1	N	
Login	Login	DTYPE_ID	15	N	
Position	Position	DTYPE_TEXT	50	N	
Position Id	PositionId	DTYPE_ID	30	N	
Position Integration Id	PositionIntegrationId	DTYPE_TEXT	30	N	
Sub Account					

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Requir ed	Notes
Account Status	AccountStatus2	DTYPE_TEXT	30	N	Bound by ACCOUNT_STATUS
Id	Id	DTYPE_ID	100	Y	
Location	Location	DTYPE_TEXT	50	N	
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40	N	
Name	Name	DTYPE_TEXT	100	Y	
Party UId	PartyUId	DTYPE_TEXT	30	Y	
Type	Type	DTYPE_TEXT	255	Y	Bound by ACCOUNT_TYPE
operation	operation	DTYPE_TEXT	30	N	
searchspec	searchspec	DTYPE_TEXT	250	N	

Additional License Restrictions A

Siebel Universal Customer Master Applications are licensed subject to the following additional license grants and restrictions. These restrictions supersede all other grants and restrictions.

Additional license grants and restrictions regarding programs of Siebel Universal Customer Master Applications. Programs of Siebel Universal Customer Master Applications are licensed to the Customer under the terms of the Agreement applicable to Programs, subject to the following additional grants and restrictions:

- Customer shall have no right to use the user interface layer of the Siebel eBusiness Application Programs with exception of the Siebel Universal Customer Master Applications administration screens;
- Customer shall have the right to access the database containing the Siebel Data Model in any way required including direct SQL access through an external application;
- Customer shall have the right to use the Siebel Data Model solely as the physical Customer Master and any extensions made through the use of Siebel Tools;
- Customer shall have the right to use the Siebel Tools to configure Siebel Universal Customer Master Applications solely in accordance with the Documentation to configure only those Business Objects and Business Components (BusComps) included with the licensed Siebel Universal Customer Master Applications through the Siebel Object Manager and Siebel eBusiness Application Integration (Siebel eAI), Siebel Enterprise Integration Manager (Siebel EIM), and Siebel Business Process Designer;
- Customer shall have the right to use System Administration solely in accordance with the Documentation including Siebel Business Process Designer, Siebel EIM administration, and Siebel eAI configuration;

- Customer shall have the right to have programmatic access only to those Business Objects and BusComps included with the licensed Siebel Universal Customer Master Applications, solely through Siebel Object Manager and Siebel eAI, Siebel EIM, and Siebel Business Process Designer;

- Notwithstanding anything to the contrary, Customer shall have no right to (i) use screens/views/applets or the Siebel user interface layer; (ii) extend or modify the Siebel Data Model other than through Siebel Tools; (iii) use the tables in the Siebel Data Model for any Siebel Programs not licensed by Customer (For example, Siebel PRM); (iv) use the Siebel Tools for general application development purposes, or (iv) use or access the Programs of Siebel Universal Customer Master Applications (including the Siebel Data Model) for Siebel non-CRM (Customer Relationship Management) applications;

- Customer will not use this licensing model of Siebel Universal Customer Master Applications to avoid paying Program License Fees for (1) full use of the functionality contained in other Siebel programs other than the Programs of Siebel Universal Customer Master Applications, or (2) the ability to use the Programs of Siebel Universal Customer Master Applications outside of the specific license grants listed above.