



Siebel Master Data Applications Reference for Industry Applications

Version 7.7
June 2004

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404
Copyright © 2004 Siebel Systems, Inc.
All rights reserved.
Printed in the United States of America

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

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

Other product names, designations, logos, and symbols may be trademarks or registered trademarks of their respective owners.

PRODUCT MODULES AND OPTIONS. This guide contains descriptions of modules that are optional and for which you may not have purchased a license. Siebel's Sample Database also includes data related to these optional modules. As a result, your software implementation may differ from descriptions in this guide. To find out more about the modules your organization has purchased, see your corporate purchasing agent or your Siebel sales representative.

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

Proprietary Information

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

Contents

Chapter 1: What's New in This Release

Chapter 2: About Siebel Master Data Applications

About Siebel Master Data Applications Modules 8

Product Components Included with Siebel Master Data Applications 10

Chapter 3: Siebel Master Data Applications Scenarios

Siebel Universal Customer Master Scenario 1 13

Siebel Universal Customer Master Scenario 2 14

Chapter 4: Architecture and Framework

About Siebel Master Data Applications Framework and Interactions 17

About Siebel Master Data Applications Functions and Services 18

About Integration Services for Siebel Master Data Applications 20

About Siebel Master Data Applications Architecture 21

UCM User Interface Layer 22

UCM Business Objects Layer 22

UCM Data Objects Layer 30

Siebel Connector and Integration of Siebel Master Data Applications 31

Chapter 5: Siebel Master Data Applications Administration

Registering Systems Connected to Siebel Master Data Applications 47

Setting System Privileges 49

Configuring System Publish and Subscribe 50

Creating Key Maps in Siebel UCM 55

About Siebel Master Data Application Integration Objects 57

UCM Integration Objects 58

UCM Envelope Integration Objects 58

UCM Dispatcher Map Integration Objects 59

Siebel Master Data Applications Integration Services 60

Chapter 6: Siebel Master Data Applications Messages

About CRMML Envelope and Header Information 72

Universally Unique Identifier 76

Siebel Master Data Application Messages 76

Contact Profile 77

Contact Address 81

Contact Activity 84

Contact Product 88

Contact Relationship 92

Contact 97

Household 107

Personal Address 112

FINCORP Account 113

Account Profile 116

Account Address 120

Account Activity 125

Account Product 132

Account Relationship 138

Account 146

Cross Reference 156

Appendix A: Additional License Restrictions

Additional License Grants and Restrictions Regarding Siebel Master Data Applications
163

Index

1

What's New in This Release

What's New in Siebel Master Data Applications Reference for Industry Applications, Version 7.7

Table 1 lists changes described in this version of the documentation to support Release 7.7 of the software.

Table 1. New Product Features in Siebel Master Data Applications Reference for Industry Applications, Version 7.7

Topic	Description
"About Siebel Master Data Applications Modules" on page 8	Two new modules, the Siebel Universal Field Service Master and the Siebel Universal Identification Master, have been added.
"Product Components Included with Siebel Master Data Applications" on page 10	The new Entity Relationship Designer is a visual design tool that allows users to create entity relationship diagrams to represent a business.
"UCM Transaction Manager" on page 31	The UCM Transaction Manager has been enhanced with new user properties and method arguments.
"UCM Converter" on page 35	Arguments and argument specifications for the UCM Converter have been revised and enhanced.
"UCM Security Manager" on page 42	The IXMLOperation_QUERYPAGE user property has been added to the UCM Security Manager.
"Cross Reference" on page 156	Cross-reference application messages handle query and response operation regarding cross-references for account, contact, and household.

2

About Siebel Master Data Applications

Siebel Master Data 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 Universal Activity Master
- Siebel Universal Product Master
- Siebel Universal Marketing Master
- Siebel Universal Sales Master
- Siebel Universal Service Master
- Siebel Universal Field Service Master
- Siebel Universal Identification 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 Master Data Applications, see ["About Siebel Master Data Applications Modules" on page 8](#). Siebel Master Data Applications act primarily as an information database and are not intended to function as an end-user application.

When Siebel Master Data Applications are used as the master database of an organization's data, one or more of the modules—through Siebel 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 Master Data Applications include a data manager, an object manager, and a user interface for administrators included with the Data Steward licensing options.

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

- By invoking business integration processes currently available with UAN Customer Lifecycle Management. See *Siebel Business Integration Processes Guide* 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 [“About Siebel Master Data Applications Framework and Interactions”](#) on page 17 for further information.

About Siebel Master Data Applications Modules

Siebel Master Data Applications include the base module (Siebel Universal Customer Master) and seven additional modules comprising various subsets of enterprise-wide customer related data:

- [“Siebel Universal Customer Master”](#)
- [“Siebel Universal Activity Master”](#)
- [“Siebel Universal Product Master”](#)
- [“Siebel Universal Marketing Master”](#) on page 9
- [“Siebel Universal Sales Master”](#) on page 9
- [“Siebel Universal Service Master”](#) on page 9
- [“Siebel Universal Field Service Master”](#) on page 9
- [“Siebel Universal Identification Master”](#) on page 10

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 Universal Activity Master

The Siebel Universal Activity Master module is an application extension to the Siebel Universal Customer Master 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 Universal Product Master

The Siebel Universal Product Master module is an application extension to the Siebel Universal Customer Master that stores an organization's product information. This extension creates a common and consistent definition of an organization's products and catalog. The key entities included with this module are product line management, product configuration, complex products, and price list.

Siebel Universal Marketing Master

The Siebel Universal Marketing Master module is an application extension to the Siebel Universal Customer Master that stores an organization's marketing information. This extension creates a central storage location for marketing efforts across an organization's multiple channels and disparate systems. The key entities included with this module are campaigns, offers, and responses.

Siebel Universal Sales Master

The Siebel Universal Sales Master module is an application extension to the Siebel Universal Customer Master that stores an organization's sales information. This extension creates a central storage location for sales efforts across an organization's multiple channels and disparate systems. The key entities included with this module are opportunities, opportunity products, competitors, decision issues, quotes, and orders.

Siebel Universal Service Master

The Siebel Universal Service Master module is an application extension to the Siebel Universal Customer Master that stores an organization's service information. This extension creates a central storage location for service efforts across an organization's multiple channels and disparate systems. The key entities included with this module are service requests, solutions, and account and policy details.

Siebel Universal Field Service Master

The Siebel Universal Field Service Master module is an application extension to the Siebel Universal Customer Master that stores an organization's field service information. This extension creates a central storage location for field service efforts across an organization's multiple channels and disparate systems. The key entities included with this module are field service activities, service agreements, asset management, invoices, inventory, repair, scheduling, and preventive maintenance.

Siebel Universal Identification Master

The Siebel Universal Identification Master module is an application extension to the Siebel Universal Customer Master that provides a secure repository for customers' social security numbers and generates a new Unique Universal ID cross-referenced with the customer's social security number. Siebel Universal Identification Master allows companies to comply with new privacy legislation that limits the use of the social security number as a unique customer identifier. The legislation requires companies to safeguard the social security numbers of their customers and severely limits the use of social security numbers in customer interactions.

Product Components Included with Siebel Master Data Applications

Certain product components of Siebel eBusiness Applications are included with Siebel Master Data Applications and provide, or add, functionality to the master files. Only the Siebel products required to run Siebel Master Data Applications are included with the Siebel Master Data Applications base product. The following eBusiness Application products are included with Siebel Master Data 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 System 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.

- Siebel Entity Relationship Designer

The Entity Relationship Designer is a visual design tool that allows you to create entity relationship diagrams (ERDs) to represent your business and then map the entities and relationships depicted in the diagram to objects in the Siebel repository, such as business components, links, and joins. See *Configuring Siebel eBusiness Applications* for further information.

Based on the software purchased, one or more of the following Siebel products and functionality may also be relevant to Siebel Master Data Applications implementation:

■ 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 development environment for configuring/customizing various aspects of Siebel eBusiness Applications, including Data Layer, Object Layer, User Interface Layer, and publish/subscribe services. See *Configuring Siebel eBusiness Applications* 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 *Applications Administration Guide* and *Siebel Employee Relationship Management Administration Guide* for further information.

3

Siebel Master Data Applications Scenarios

This chapter examines a fictional financial institution, its business requirements, and the Siebel Master Data 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 Master Data Applications, see [Chapter 2, "About Siebel Master Data Applications."](#)

Siebel Universal Customer Master Scenario 1

The following business scenario is designed to provide guidance for a potential Siebel Master Data Applications deployment. The business institution and its existing implementation are described, followed by the benefits of the Siebel Master Data 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.

The solution is to continue to use the Siebel Master Data Applications as a key component in the synchronization of customer profile information across customer-related applications, including back-office applications.

Siebel Master Data Applications Deployment

In this scenario, Siebel Master Data Applications operate as a stand-alone application acting without a Customer Relationship Management (CRM) deployment in an application network. Siebel Master Data 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.

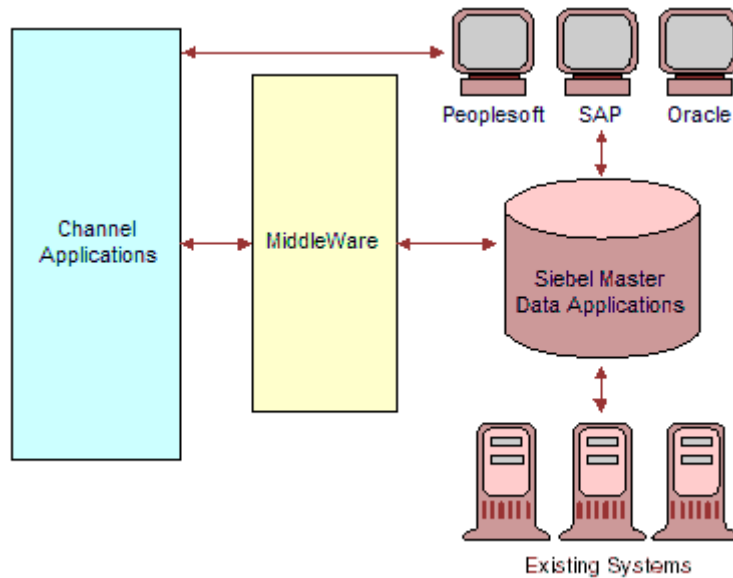


Figure 1. Siebel Master Data Applications Stand-Alone Deployment Scenario

Siebel Master Data Applications Integration

The Siebel Master Data 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 Master Data Applications"](#) on page 31 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 Master Data Applications deployment changes in this scenario, but the business solution provided by Siebel Universal Customer Master remains the same.

Siebel Master Data Applications Deployment

In this scenario, if the Siebel Master Data Applications deployment works with one Siebel eBusiness Application, as well as with other back-office applications, the Siebel eBusiness Application relies on Siebel Master Data Applications as the Siebel database. Siebel Master Data Applications use middleware infrastructure to consolidate other customer data from back-office systems. See [Figure 2](#).

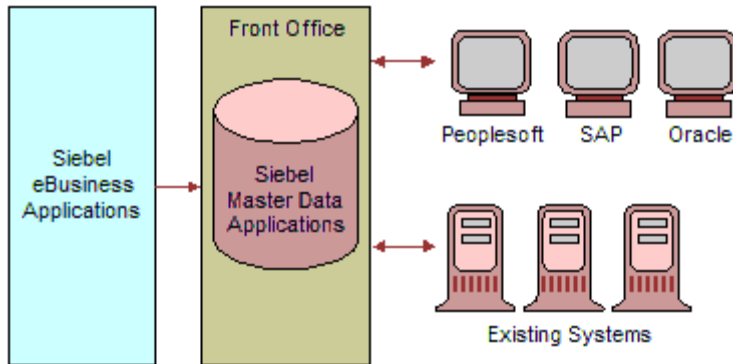


Figure 2. Siebel Master Data Applications Deployed with a Siebel eBusiness Application

Alternatively, if Siebel Master Data Applications work in a deployment including multiple instances of Siebel eBusiness Applications, as well as other back-office systems, Siebel Master Data Applications act as the central database of enterprise information for all systems. See [Figure 3](#).

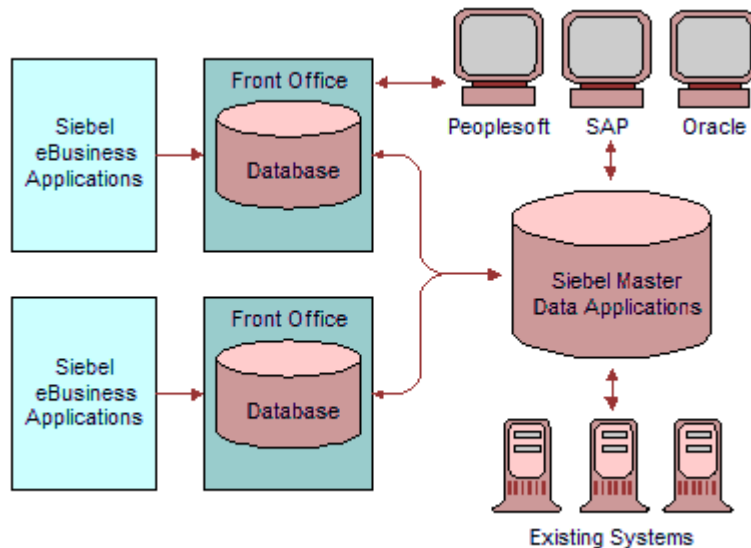


Figure 3. Siebel Master Data Applications Deployed with a Multi-Siebel eBusiness Application Environment

Siebel Master Data Applications Integration

As in the previous scenario, the Siebel Master Data 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 Master Data Applications. See [“Siebel Connector and Integration of Siebel Master Data Applications” on page 31](#) for further information.

4

Architecture and Framework

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

About Siebel Master Data Applications Framework and Interactions

Siebel Master Data 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 Master Data Applications and Siebel eBusiness Applications or back-office applications using enterprise integration and the regular links to Siebel eBusiness Applications. See [Figure 4](#) for this relationship.

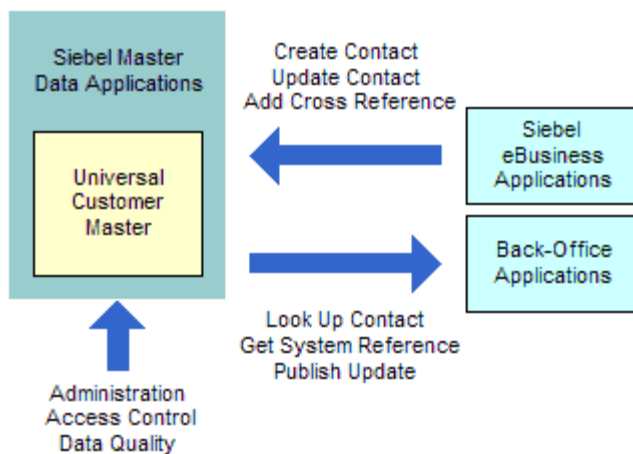


Figure 4. Siebel Master Data Applications Framework

- For information about the functionality included with Siebel Master Data Applications, see ["About Siebel Master Data Applications Functions and Services"](#) on page 18.
- For information about the administrative functionality of Siebel Master Data Applications, see ["Administering Siebel Master Data Applications"](#) on page 19.
- For information about the technical integration aspects used with Siebel Master Data Applications, see ["Siebel Connector and Integration of Siebel Master Data Applications"](#) on page 31.

About Siebel Master Data Applications Functions and Services

The database tables and Siebel Connector for Master Data Applications support insert, delete, update, and query functionality. When 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 Master Data Applications. See [Chapter 6, "Siebel Master Data Applications Messages,"](#) for details about prebuilt UCM services interfaces for accessing these functions.

Siebel Master Data Applications also provide the following services through configuration:

- **Publish.** Publish data records from Siebel Master Data Applications.
- **Request and Reply.** Request data from Siebel Master Data Applications and return data.

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

Siebel Master Data Applications use a variety of integration techniques to allow the operation of functions and services between other applications and themselves. See ["About Siebel Master Data Applications Architecture" on page 21](#) for further information.

Administering Siebel Master Data Applications

Siebel Master Data Applications are administered through Siebel eBusiness Application user interface screens. However, only those screens necessary for viewing and administering the information contained within Siebel Master Data Applications are included. See [Chapter 5, "Siebel Master Data 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](#).

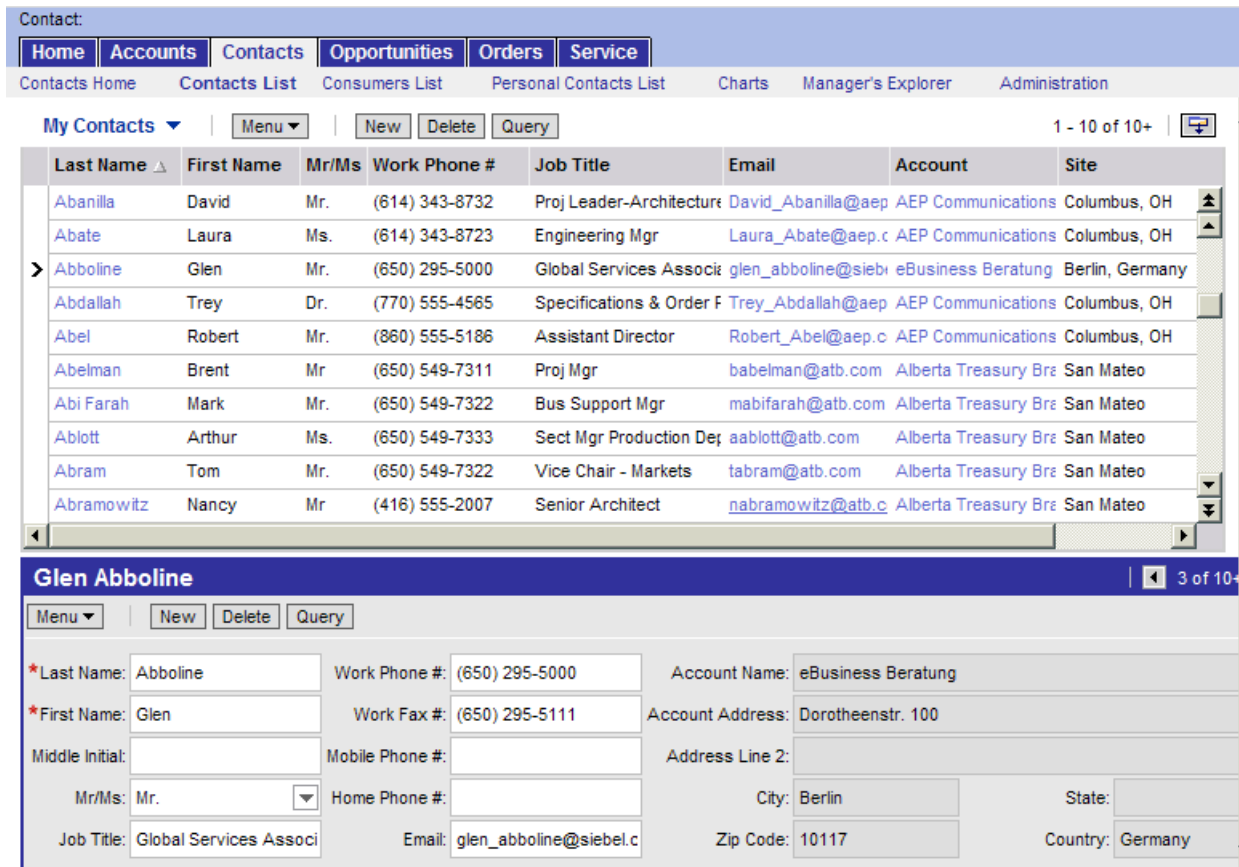


Figure 5. Siebel eBusiness Applications Contacts Screen

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

Access Control is enforced at the user interface level for Siebel Master Data Applications administration screens. Siebel Universal Customer Master Application messages have full visibility of the data, that is no access control mechanism is applied to the messages but they will be subject to system privilege checks when accessing the Universal Customer Master Application. For further information and procedures, see *Configuring Siebel eBusiness Applications* and *Security Guide for Siebel eBusiness Applications*.

About Integration Services for Siebel Master Data Applications

The Siebel Connector for Master Data Applications provides integration between Siebel Master Data Applications and other XML-based applications that support Siebel Universal Customer Master Application business process specifications. The Siebel Connector for Master Data 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 Master Data 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 Master Data 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 Master Data 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 3, "Siebel Master Data Applications Scenarios"](#) for scenarios using the Siebel Master Data Applications.

In addition, the Siebel Connector for Master Data 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, for more information see ["Creating Key Maps in Siebel UCM"](#) on page 55.
- 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.

About Siebel Master Data Applications Architecture

Siebel Master Data 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.

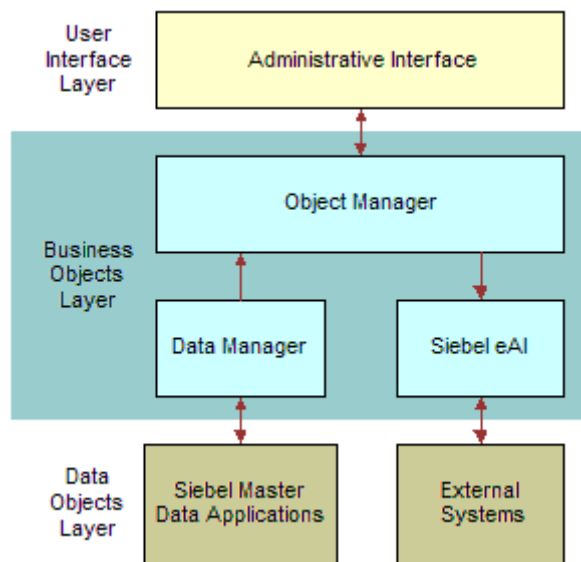


Figure 6. Siebel Master Data Applications n-tier Architecture

The Siebel Master Data 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 current tasks or client sessions.
- Supports multithreaded and interactive components, except background mode components.

UCM User Interface Layer

The user interface layer for Siebel Master Data 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 *Configuring Siebel eBusiness Applications*. The user interface layer is used for the administration of the Siebel eBusiness Information suite; see ["Administering Siebel Master Data Applications" on page 19](#) for further information.

UCM Business Objects Layer

The business logic layer for Siebel Master Data 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 and how to customize it, see *Configuring Siebel eBusiness Applications*.

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

Object Manager and Data Manager

The object manager and data manager layer use:

- Business Objects
- Business Components
- Virtual Business Components
- External Business Components

These business objects are customizable. For further information about the business objects and data objects layer, see *Configuring Siebel eBusiness Applications*.

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 Master Data 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.

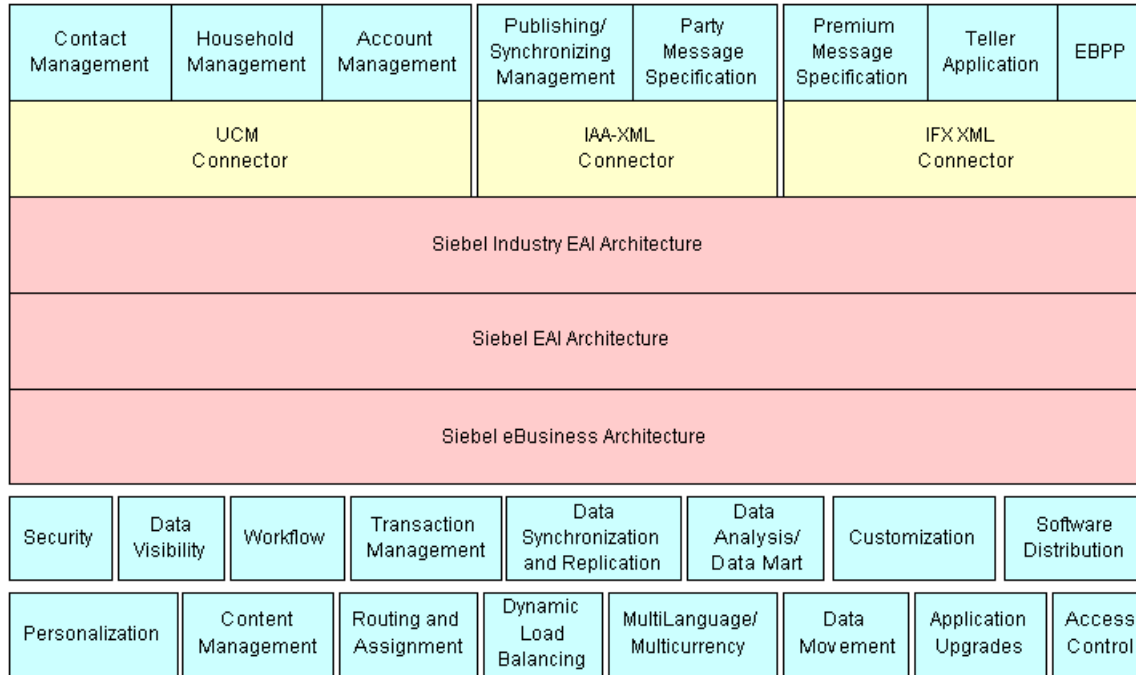


Figure 7. High-Level Architecture of Siebel Industry eAI and Standard Connectors

Siebel Connector for Master Data Applications is based on Siebel XML standards 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 Master Data 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 Master Data Applications includes the UCM Dispatcher, UCM Converter, UCM Publish/Subscribe Service, UCM Security Manager, and UCM Transaction Manager.Master Data.

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 Master Data Applications such as <InsertContactProduct>, <DeleteAccountAddress>, <UpdateContactProduct>, <UpdateHousehold>, <DeleteAccountAddressResponse>, <LookUpAccount>, <LookUpPersonalAddressByState>, and so on. These messages are included in Siebel Master Data 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.

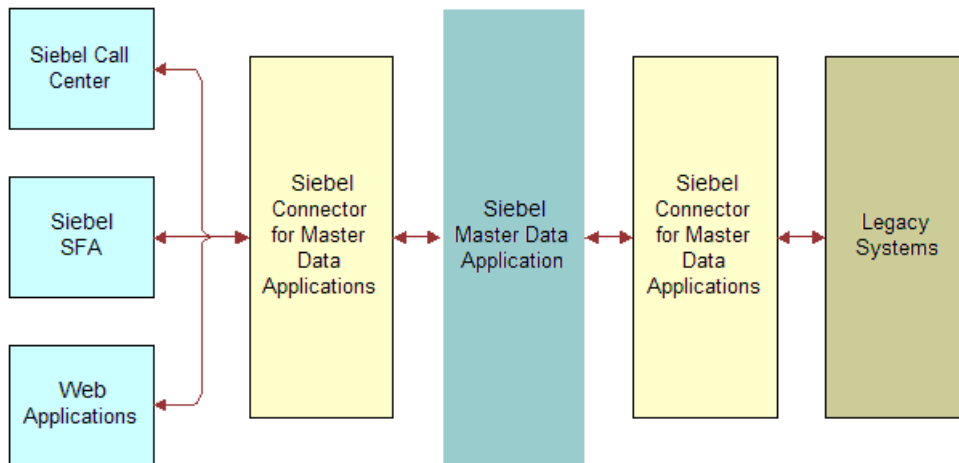


Figure 8. Business Process Flow

Inbound Data Flow

As illustrated in Figure 8, 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 Master Data 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 UCM Dispatcher traverses the XML instance and identifies the messages received according to the rule sets of the UCM Dispatcher Map. It then identifies the envelope, header, and body sections among the hierarchy nodes and sends it to the UCM Security Manager.

NOTE: The Dispatcher Map is shipped as a part of Siebel UCM Applications. For details, see “UCM Dispatcher Map Integration Objects” on page 59.

- 3 The UCM Security Manager takes the XML instance, verifies the System for registration with Siebel Master Data 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 UCM Converter.
- 4 The UCM 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 UCM Transaction Manager.
- 5 The UCM 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 UCM 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 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 Master Data Applications Integration. Supporting integration objects for Siebel Master Data Applications business processes are shipped as part of the Siebel Master Data 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 UCM Transaction Manager, the EAI Siebel Adapter, the UCM 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 UCM 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 UCM 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 Master Data Applications System. Sender plays an insurance application role and receiver plays a 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 the creation time of a message and <TimeToLive> is the expiration time of a message. Receiver should disregard a message if the message has 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>
    </Contact>
  </InsertContactProfile>
</CIFPartyPackage>

```

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 UCM 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 Master Data Applications.

```

<CRMML>
  <CRMHeader>
    ...
  </CRMHeader>

  <CIFPartyPackage ID="p1">
    <InsertContactProfile ID="m1">
      <Contact ID="a1">
        <Id>ext2223</Id>
      </Contact>
    </InsertContactProfile>
  </CIFPartyPackage>
  ...

```

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

UCM Data Objects Layer

The Data Objects layer for Siebel Master Data 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 Master Data 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 *Configuring Siebel eBusiness Applications*.

Siebel Connector and Integration of Siebel Master Data Applications

Siebel Connector for Master Data Applications describes a set of configurable components for integration among enterprise applications and Siebel Master Data Applications.

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

- UCM Transaction Manager
- UCM Converter
- UCM Dispatcher
- UCM Security Manager
- UCM Publish/Subscribe Service

UCM Transaction Manager

The UCM Transaction Manager executes operations specified in Universal Customer Master Application message instances as Siebel database transactions. The UCM 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 2 lists the user properties for UCM Transaction Manager.

Table 2. User Properties of UCM 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.
EnableAgentLock	TRUE	Enable system preference EAIAgent

Table 2. User Properties of UCM Transaction Manager

Name	Value	Description
IXMLOperation_ADD	EAI Siebel Adapter/Insert/	Predefined operation
IXMLOperation_DELETE	EAI Siebel Adapter/Delete/	Predefined operation
IXMLOperation_QUERY	EAI Siebel Adapter/Query/	Predefined operation
IXMLOperation_QUERY PAGE	EAI Siebel Adapter/QueryPage/ NewQuery=>IXML_NewQuery;SearchSpec=>SearchSpec;StartRowNum=>IXML_StartRowNum;PageSize=>IXML_PageSize;SERVICE_VALUE_RETURN;\$LastPage=>IXML_LastPage;\$NumOutputObjects=>IXML_NumOutputObjects;	Predefined operation
IXMLOperation_QUERY_BY_ID	EAI Siebel Adapter/Query/ PrimaryRowId;!SiebelMessage;	Predefined operation
IXMLOperation_SYNCH	EAI Siebel Adapter/Synchronize/	Predefined operation
IXMLOperation_UPDATE	EAI Siebel Adapter/Update/	Predefined operation

The UCM Transaction Manager uses the prebuilt methods described in [Table 3](#) to process inbound messages.

Table 3. Pre-built Methods for UCM Transaction Manager

Name	Value	Description
Execute	Execute Transaction	Executes the transaction by invoking predefined business services
ExecuteOutbound	Execute Outbound	Executes the outgoing transaction
ExecuteSave	Execute and Save	Executes the transaction and saves result into memory

UCM Transaction Manager business service uses combinations of user property, method and method arguments to achieve different tasks. [Table 4](#) lists the method arguments available to use with UCM Transaction Manager business service.

Table 4. Method Arguments for Use with UCM Transaction Manager

Argument	Default Value	Description
OnlyIOI	FALSE	Used for Inbound integration. The inbound message may contain header, body, and envelope sections. When the Transaction Manager takes the proper operation against the Siebel application, the integration object instances for response are generated as well. Information from the request message is dropped if this value is set to true, reducing overhead for the UCM Converter and others. If this value is not set to true, the request information is still carried over.
ReportErrInMsg	FALSE	If set to true, the method will try to generate an error object within the message (default is false).
RollbackInError	FALSE	If set to true, the method will try to trigger a rollback operation when an error is encountered (default is false).
XMLHierarchy	N/A	Property set in UCM integration object hierarchy.
IXMLMapPath		The dispatch map entry for the current transaction.
PrimaryRowId		The primary row Id for the integration object.
SearchSpec		Run time search specification.
SiebelFINSOperationOut		Operation name, defined in the IAA transaction manager user property.
PlaceToSave		Specifies whether the result should be saved into memory or into a file. Valid value is File or Mem.

Although these arguments are available for use by UCM Transaction Manager, not all can be used with each method. [Table 5](#) lists the arguments for each method.

Table 5. Execute Method Arguments

Name	Display Name	Data Type	Type	Optional	Comments
Method: Execute					
OnlyIOI	Produce only Integration Object Instance	String	Input	Y	If set to true, the method will take out all headers and envelope information and return only integration object instances.
ReportErrInMsg	Report Error In Message	String	Input	Y	If set to true, the method will try to generate an error object within the message (default is false).
RollbackInError	Rollback In Error	String	Input	Y	If set to true, the method will try to trigger a rollback operation when an error is encountered (default is false).
XMLHierarchy	XML Property Set	Hierarchy	Input / Output	Y	XML Message Hierarchy
Method: Execute Outbound					
IXMLMapPath	IXML Map Path	String	Input	N	Dispatch map entry for the current transaction.
PrimaryRowId	Primary Row Id	String	Input	Y	Primary row Id for the integration object.
SearchSpec	Search Specification	String	Input	Y	Run time search specification.
SiebelFINS OperationOut	Outbound Operation	String	Input	N	Operation name, defined in the IAA transaction manager user property.
XMLHierarchy	XML Hierarchy	Hierarchy	Output	N	XML Message Hierarchy
Method: Execute Save					
IXMLMapPath		String	Input	N	Dispatch map entry for the current transaction.

Table 5. Execute Method Arguments

Name	Display Name	Data Type	Type	Optional	Comments
PlaceToSave	Save into File/Mem	String	Input	Y	Specifies whether the result should be saved into memory or into a file. Valid value is File or Mem.
PrimaryRowId		String	Input	Y	Primary row Id for the integration object.
SearchSpec		String	Input	Y	Run time search specification.

UCM Converter

The UCM Converter converts Siebel Master Data Applications messages received to a property set for inbound communication. The converter iterates through the requests and responses in the message to construct error information (if any), and also constructs the envelope and header section of the message. [Table 6](#) lists the user properties for this converter.

Table 6. User Properties for UCM 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.
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 Master Data Applications Envelope.

This service provides several ready-to-use methods as described in [Table 7](#).

Table 7. Methods for UCM Converter

Name	Display Name	Comments
GenerateErrorPropSet	GenerateErrorPropSet	Constructs the XML hierarchy to respond to external systems when undesired operation or configuration errors occur.
PropSetToXML	PropSetToXML	Converts an integration object hierarchy to an XML document.
PropSetToXMLPropSet	PropSetToXMLPropSet	Converts an integration object hierarchy to an XML hierarchy.
XMLPropSetToPropSet	XMLPropSetToPropSet	Converts an XML hierarchy to an integration object hierarchy.
XMLToPropSet	XMLToPropSet	Converts an XML document stored in the value member of the input property set to an integration object hierarchy.

For each method defined, the UCM Converter business service takes input arguments and produces output arguments. These arguments are described in [Table 8](#).

Table 8. Arguments for the UCM Converter

Name	Default	Description
ErrorCode	N/A	Actual error code to be set to the <ErrorId> element in CRMML fault section.
ErrorMessageText	N/A	Actual error text to be set to the <ErrorDescription> element in CRMML fault section.
GeneralErrorMessageText	N/A	Generic text to be prepended to the error text.
XMLHierarchy	N/A	Property Set in the external integration object or XML hierarchy used as input/output.
<Value>	N/A	Output XML document.
EscapeNames	N/A	Escape characters not valid for XML tags. Otherwise, remove invalid characters.
IgnoreCharSetConvErrors	N/A	If some characters cannot be represented in the destination character set (for example, local codepage), the errors can be ignored. By default, the errors are not ignored. In both cases, a warning log entry is produced.

Table 8. Arguments for the UCM Converter

Name	Default	Description
XMLCharEncoding	N/A	XML character encoding to use in the output XML document. If the encoding is blank or not supported, an error will be produced.
XMLHeaderText	N/A	Text to prepend to the beginning of XML document data.
ExternalEntityDirectory	N/A	Location of external entity files, such as DTD files.
ValidateExternalEntity	N/A	If true, the parser will be set to validate against external metadata (for example, a DTD).
FailSecurityHierarchy	N/A	Property Set storing the operations which fail the security system check.

Each UCM Converter method uses a group of these method arguments to convert the data for the next service in the integration process. [Table 9](#) lists the arguments for each method.

Table 9. Argument Specifications

Name	Display Name	Data Type	Type	Optional	Comments
Method: GenerateErrorPropSet					
ErrorCode	ErrorCode	String	Input	N	Actual error code to be set in the IAA Fault section.
ErrorMessageText	ErroMessageText	String	Input	N	Actual error text to be set in the IAA Fault section.
GeneralError MessageText	GeneralError MessageText	String	Input	Y	Generic text to be prepended to the error text.
XMLHierarchy	ErrXMLHierarchy	Hierarchy	Input/ Output	Y	Input request hierarchy/output response hierarchy.
Method: PropSetToXML					
<Value>	XML Document	String	Output	Y	Output XML document.

Table 9. Argument Specifications

Name	Display Name	Data Type	Type	Optional	Comments
EscapeNames	Escape Names	String	Input	Y	Escape characters not valid for XML tags. Otherwise, remove invalid characters.
IgnoreCharSet ConvErrors	Ignore Character Set Conversion Errors	String	Input	Y	If some characters cannot be represented in destination character set (for example, local codepage), the errors can be ignored. By default, the errors are not ignored. In both cases, a warning log entry is produced.
XMLCharEncoding	XML Character Encoding	String	Input	Y	XML character encoding to use in the output XML document. If encoding is blank or not supported, an error will be produced.
XMLHeaderText	XML Header Text	String	Input	Y	Text to prepend to beginning of the XML document data.
XMLHierarchy	XML Hierarchy	Hierarchy	Input	N	XML Property Set hierarchy. Document property set is a child of XML Property Set child of the Service Arguments.
FailSecurityHierarchy	FailSecurityHierarchy	Hierarchy	Input	Y	

Table 9. Argument Specifications

Name	Display Name	Data Type	Type	Optional	Comments
Method: PropSetToXMLPropSet					
XMLHierarchy	XML Hierarchy	Hierarchy	Input	N	XML Property Set hierarchy. Document property set is a child of XML Property Set child of the Service Arguments.
Method: XMLToPropSet					
<Value>	XML Document	String	Input	N	Input XML document.
EscapeNames	Escape Names	String	Input	Y	XML Tags may contain names that have invalid characters escaped. Character escape format is internal to Siebel.
ExternalEntity Directory	External Entity Directory	String	Input	Y	Location of external entity files, such as DTD files.
IgnoreCharSet ConvErrors	Ignore Character Set Conversion Errors	String	Input	Y	If some characters cannot be represented in destination character set (for example, local codepage), the errors can be ignored. By default, the errors are not ignored. In both cases, a warning log entry is produced.

Table 9. Argument Specifications

Name	Display Name	Data Type	Type	Optional	Comments
ValidateExternal Entity	Validate External Entity	String	Input	Y	If true, the parser will be set to validate against external metadata (for example, a DTD).
XMLCharEncoding	XML Character Encoding	String	Output	Y	Character encoding of the XML document, detected by the converter independent of the parser.
XMLHierarchy	XML Hierarchy	Hierarchy	Output	Y	Output XML Property Set Hierarchy. Document property set is a child of the XML Property Set child of Service Argument output.

UCM Dispatcher

The UCM 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 10](#) lists the user properties for UCM Dispatcher.

Table 10. User Properties for UCM 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 "UCM Envelope."

The UCM Dispatcher business service provides ready-to-use methods and method arguments described in [Table 11](#).

Table 11. UCMDispatcher 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 UCM Dispatcher business service takes input arguments and produces output arguments. These arguments are described in [Table 12](#) and [Table 13](#).

Table 12. UCM Dispatcher Method Arguments

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

Table 13. Method Arguments for DispatchMessage Method

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

UCM Security Manager

The UCM 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 UCM Security Manager cycles through the heterogeneous commands in the body. The UCM Security Manager checks the privilege of the individual operations in the body sections against the privileges in the Customer Master for the particular object and operation. Any fail operation instances in the body are removed from the XMLHierarchy and attached to the FailSecurityHierarchy. [Table 14](#) lists the user properties for the UCM Security Manager.

Table 14. User Properties for UCM Security Manager

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

Table 14. User Properties for UCM Security Manager

Name	Value	Description
IXMLOperation_QUERY PAGE	Query	Field key value pair. Checks the Query field for privilege if the UCM Dispatcher tags the business process as an IXMLOperation_QUERYPAGE operation.
IXMLOperation_UPDATE	Update	Field key value pair. Checks the Update field for privilege if the UCM Dispatcher tags the business process as an IXMLOperation_UPDATE operation.

The UCM Security Manager processes only inbound messages. [Table 15](#) lists and describes the method that processes inbound messages.

Table 15. Process Method for UCM Security Manager

Name	Display Name	Description
SecurityCheck	Security Check	Supports only inbound requests to the Siebel Connector for Master Data 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.

UCM Security Manager business service uses a combination of user property and method arguments for its methods to achieve different tasks. [Table 16](#) lists the method arguments available to use with UCM Security Manager business service.

Table 16. Method Arguments for Use with UCM 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 UCM Security Manager will bypass the registration verification process and the operation privilege checking process. UCM Security Manager will perform its security checking operations only when the value is set to true.
XMLHierarchy	N/A	Property set in UCM integration object hierarchy.
FailSecurityHierarchy	N/A	Property set in Siebel Master Data 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 17 lists the detailed arguments for the method Security Check.

Table 17. Arguments for Method Security Check

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

UCM Publish/Subscribe Service

The UCM Publish/Subscribe Service supports both Real Time and Daily Batch publish types.

The Real Time publish module takes the Siebel Connector for Master Data 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 Master Data Applications message specification and sends them 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 UCM Publish/Subscribe Service has no user properties. It uses the prebuilt methods described in Table 18 to publish messages.

Table 18. Method for UCM Publish/Subscribe

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

UCM Publish/Subscribe Service uses a combination of methods and method arguments to achieve different tasks. [Table 19](#) lists the method arguments available to use with UCM Publish/Subscribe Service business service.

Table 19. Method Arguments for Use with UCM Publish/Subscribe

Argument	Display Name	Description
XMLHierarchy	N/A	Property set in UCM 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 UCM Publish/Subscribe Service, not all can be used with each method. [Table 20](#) and [Table 21](#) list the arguments for each method.

Table 20. Argument Details for Publish Method (Real Time)

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

Table 21. 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 Master Data 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 Master Data Applications

Siebel EIM is an important component of Siebel Master Data Applications. It bulk imports and exports data residing in other back-office systems into Siebel Master Data 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*.

5

Siebel Master Data Applications Administration

This chapter describes the administrative capabilities of Siebel Master Data Applications. Only users with administration privileges have access to these screens and procedures.

Registering Systems Connected to Siebel Master Data Applications

Every application or system that connects to Siebel Master Data Applications must register through the System Registrations view of the Administration - Universal Customer Master screen. Once registered, the system's privileges and accessibility to the Siebel Master Data Applications is administered on other tabs within this view. See ["Setting System Privileges" on page 49](#) and ["Configuring System Publish and Subscribe" on page 50](#) for this information.

To create a new system

- 1 Navigate to the Administration - Universal Customer Master screen.
- 2 In the System Registrations view, click New.
- 3 Enter the appropriate information in the available fields to define the system.
Use [Table 22](#) for information on each of the fields.

Table 22. System Fields

Field	Description
System Number	The System Number represents an identifier for an application that accesses Siebel Master Data 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 Master Data Applications.
Protocol Type	The protocol that an application uses to access Siebel Master Data Applications. Currently, this value is HTTP or MQSeries.
Queue Manager Name	Name of the queue manager that receives and sends messages from Siebel Master Data 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.

Table 22. System Fields

Field	Description
URL	URL destination for posting responses to messages. Only applicable when protocol type is HTTP.
Description	Description of the application accessing Siebel Master Data Applications.
Comment	Comments regarding the application accessing Siebel Master Data Applications.

Figure 9 shows the creation of an example system.

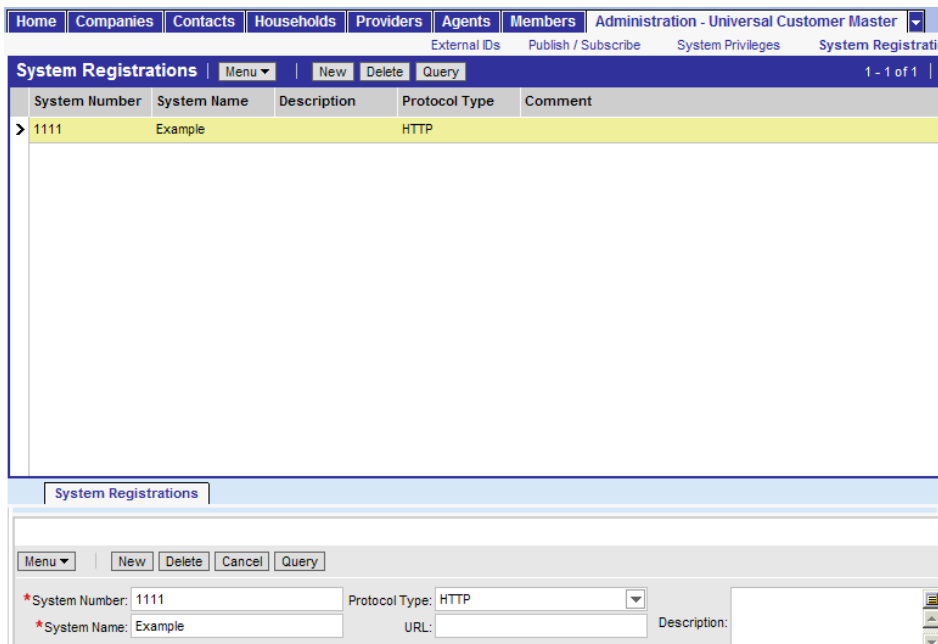


Figure 9. Creating a New System

To delete a system

- 1 Navigate to the Administration - Universal Customer Master screen.
- 2 In the System Registrations list, select the system of interest.

The System Registrations form for this system, which appears under the list, provides detailed information on this system.

- 3 In the System Registrations form, click Delete.
A confirmation dialog appears.
- 4 Select OK to delete the system.

Setting System Privileges

After a system is registered (see [“Registering Systems Connected to Siebel Master Data Applications” on page 47](#) for information on registering a system), administrators can set privileges for this system, which allows grant access to Siebel Master Data 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 Administration - Universal Customer Master screen.
- 2 In the Systems Registrations list, select the system of interest.
- 3 From the link bar, select System Privileges.
The System Detail form and System Privileges view appear.
- 4 In the System Privileges view, click New.
- 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 Master Data Applications like 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 “Configuring System Publish and Subscribe” on page 50 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.

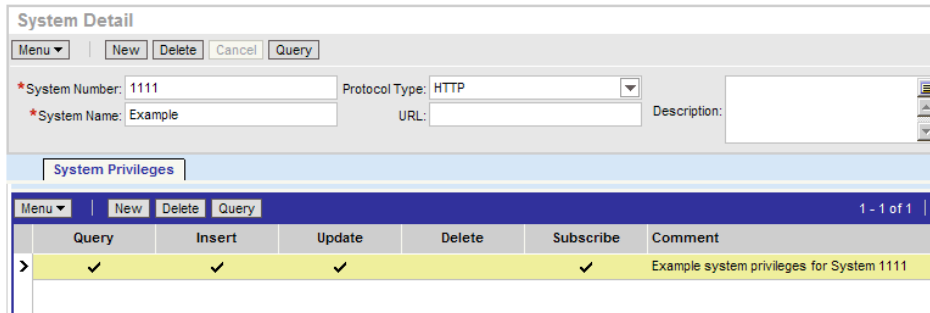


Figure 10. Defining System Privileges

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 “Configuring System Publish and Subscribe” on page 50 for further information.

Configuring 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 flag 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 “Setting System Privileges” on page 49 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 52).

To configure real-time publish and subscribe for a system

- 1 Navigate to the Administration - Universal Customer Master screen.
- 2 In the System Registrations list, select the system of interest.

- 3** From the link bar, select Publish/Subscribe.

The System Detail form and Publish/Subscribe view 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 Master Data Applications like Contact, Account, Household, and so on.
Publish Frequency	How often the Siebel Master Data 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 52 for the daily batch procedure.
Last Published	The last time the Siebel Master Data Applications published 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 UCM 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 UCM 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.

The screenshot shows the 'System Detail' form in Siebel. The top section contains fields for System Number (1111), System Name (Example), Protocol Type (MQSeries), Queue Manager Name, and Queue Receiver Channel. Below this is a 'Publish / Subscribe' button. The bottom section is a table with columns: Object Name, Publish Frequency, Publish Time, Last Published, Start Date, End Date, and Comment. A single row is visible for the 'Account' object, with a 'Real Time' publish frequency and a comment 'Example system privileges'.

Figure 11. Defining System Real-time Publish and Subscribe Capabilities

To configure batch-time publish and subscribe for a system

- 1 Navigate to the Administration - Universal Customer Master screen.
- 2 In the System Registrations list, select the system of interest.
- 3 From the link bar, select Publish/Subscribe.

The System Detail form and Publish/Subscribe view 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 Master Data Applications like Contact, Account, Household, and so on.
Publish Frequency	How often the Siebel Master Data 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 50 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 Master Data Applications published 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 From the menu drop-down list, select New Record. Create a new workflow process with any name, such as UCM Account Dailybatch Publish Workflow, as in the following figure:



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

- 7 Configure the CIFAccountDailybatch module as follows:

Step	Start
Step	UCM Account Dailybatch
Business Service	UCM 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 UCM 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 UCM Contact messages or UCM 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 UCM Batch Workflow on a batch-time basis using the following procedure. See *Siebel System Administration Guide* for detailed information on this process.
 - a Navigate to the Administration - Server Management screen.
 - b From the link bar, click Jobs.
 - c Select the Job Detail view tab and click New.

- d** In the Job Detail form, enter the following information in the available fields:

Field	Value
Component/Job	Workflow Process Batch Manager.
Scheduled Start	Scheduled start date and time of the component job.
Expiration	Date at which the component job is no longer valid.
Requested Server	Siebel
Request Key	Set if you want to target a component or repeating component job to a specific instance of the server component identified by the request key. In all other situations, keep this field blank.
Delete Interval	Set with Delete Unit field, this field determines the length of time before the component job is deleted. If not updated, this field defaults to 1.
Delete Unit	Set with Delete Interval field, this field determines the length of time before the component job is deleted. If not updated, this field defaults to Weeks.
Repeating?	Check this box to make the component job a repeating component job.
Repeat Unit	Set with Repeat Interval field, this field determines the length of time before the component job repeats.
Repeat Interval	Set with Repeat Unit field, this field determines the length of time before the component job repeats.
Repeat From	Actual Start.
Repetitions	Number of times the component job repeats.

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

- e** In the Job Parameters form, click New.
f 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 UCM Account Dailybatch Publish Workflow

Figure 12 displays the definition of batch-time publish and subscribe capabilities for the example system.

Object Name	Publish Frequency	Publish Time	Last Published	Start Date	End Date	Comment
> Account	Daily Batch	12:00AM	3/10/2004 11:44:14	3/15/2004 11:45:04	3/31/2004 11:45:10	Example system privileges

Figure 12. Defining System Batch-time Publish and Subscribe Capabilities

Creating Key Maps in Siebel UCM

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-many mapping of customer records across multiple systems throughout the organization. In other words, one record in Siebel UCM can map to one or many records in any registered applications in UCM. Siebel Master Data 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 Master Data 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 Master Data Applications Administration Screen.

To create a new Account Reference record

- 1** Navigate to the Administration - Universal Customer Master screen.
- 2** From the link bar, select External IDs.
The Accounts list and External Account IDs view appear.
- 3** In the Accounts list, select the record of interest.
- 4** In the External Account IDs view, click New.

- 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 Master Data 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 Master Data Applications. This number is picked from the Systems entered in the System Registrations 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 Identifier of the external system record. This value is the only field stored in the <Id> tag in request message.
External Id2	The External Id2 stores an optional second identifier of the external system record.
External Id3	The External Id3 stores an optional third identifier of the external system record.
Comment	Comments regarding the external system record referencing the record in the Siebel Universal Customer Master Application.

Figure 13 displays the creation of an example Account reference record.

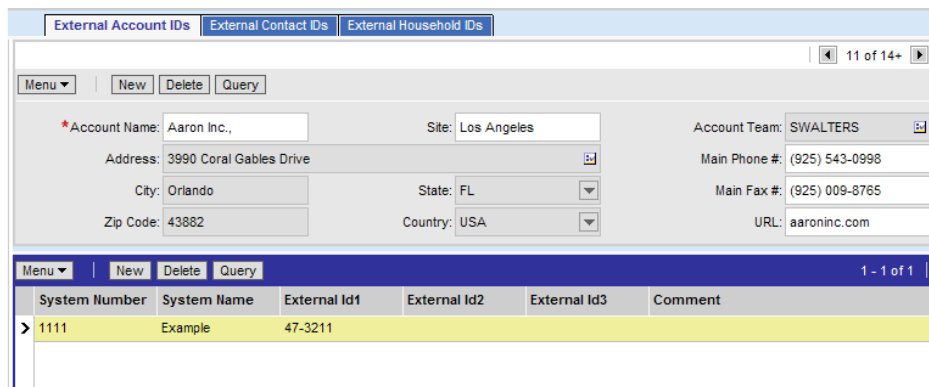


Figure 13. Creation of an Example Account Reference Record

To create a new Contact Reference record

- 1 Navigate to the Administration - Universal Customer Master screen.
- 2 From the link bar, select External IDs.

- 3** Select the External Contact IDs view tab.
The Contacts list and External Contact IDs view appear.
- 4** In the Contacts list, select the record of interest.
- 5** In the External Contact IDs view, click New.
- 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 56](#) for information on each of these fields.

To create a new Household Reference record

- 1** Navigate to the Administration - Universal Customer Master screen.
- 2** From the link bar, select External IDs.
- 3** Select the External Household IDs view tab.
The Households list and External Household IDs view appear.
- 4** In the Households list, select the record of interest.
- 5** In the External Household IDs view, click New.
- 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 56](#) for information on each of these fields.

About Siebel Master Data Application Integration Objects

There are three types of Siebel Master Data Applications integration objects:

- "UCM Integration Objects"
- "UCM Envelope Integration Objects"
- "UCM Dispatcher Map Integration Objects"

UCM Integration Objects

The UCM integration object contains 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 *Integration Platform Technologies: Siebel eBusiness Application Integration Volume II* for further information. This internal integration object is required so that the UCM Transaction Manager can 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 UCM Integration Objects

- 1 Determine whether or not you need to customize any integration objects listed in [Table 30 on page 67](#), 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 UCM 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 6, "Siebel Master Data Applications Messages."](#)

UCM 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 UCM Envelope.

To modify UCM 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, see ["CRMML Standard" on page 26](#) and ["About CRMML Envelope and Header Information" on page 72](#).

UCM Dispatcher Map Integration Objects

The UCM dispatcher map is an integration object that contains the rule sets used by the UCM Dispatcher. The default UCM dispatcher map is CIFDispMap. Siebel Master Data 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

For performance reasons in a production environment, deactivate business processes that you are not supporting. 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.
- 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;IXMLOperation_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 UCM 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 Master Data Applications Integration Services

Siebel Master Data 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 that you can configure to meet your business requirements:

- UCM Transaction Manager
- UCM Converter
- UCM Dispatcher
- UCM Security Manager
- UCM 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 using Siebel Tools to configure your application, see *Configuring Siebel eBusiness Applications*.

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.

UCM 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 an argument of the Connector Integration Object Instance.

Table 26 lists the user properties examples for the UCM Transaction Manager.

Table 26. User Properties for UCM 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

UCM 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 UCM Converter

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

The UCM Converter uses the hierarchy represented in the UCM integration object to guide the message through the converting process. If the integration object instance receives elements 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.

UCM Dispatcher

You can modify both user properties for this business service as shown in [Table 28](#).

Table 28. User Properties for UCM 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.

UCM Security Manager

You can modify or extend this business service by manipulating its user properties. The Name column stands for the UCM 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 UCM Security Manager

Name	Value
IXMLOperation_ADD	Insert
IXMLOperation_DELETE	Delete
IXMLOperation_QUERY	Query
IXMLOperation_QUERYPAGE	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 job, which requests one or more server tasks to run. The server task you need to run for the Siebel Connector for Master Data 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 Administration - Server Configuration screen.
- 2 From the link bar, click Enterprises.
- 3 Select the Profile Configuration view tab.
- 4 Create two new records in the Profile Configuration list and provide the following information.

	Record 1	Record 2
Name	Any name, CIFMQConnSubsy for example	Any name, CIFMQDataSubsys for example
Alias	This field is required for creation of this record.	This field is required for creation of this record.
Subsystem Type	MQSeriesServerSubsys	EAITransportDataHandlingSubsys

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

- 5 For each record, modify the following parameters in the Profile Parameters list:

Parameter Name	Record 1	Record 2
MQSeries Physical Queue Name	Queue name to receive inbound request message from	N/A
MQSeries Queue Manager Name	Queue manager name who own the queues	N/A
MQSeries Response Physical Queue Name	Queue name to send response message to	N/A
MQSeries Sleep Time	100 (or longer if required)	N/A
Workflow Process to Execute	N/A	UCM Server Party Package Workflow (MQSeries Server Transport Error Handling)

- 6 Save both records.

For details on this procedure and the Named Subsystems, see *Siebel System Administration Guide*.

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

To configure MQSeries Server Receiver parameters

- 1 Navigate to the Administration - Server Configuration screen.
- 2 From the link bar, click Servers.
- 3 In the Siebel Servers list, select the Siebel Server of interest.
- 4 Click the Components view tab.
- 5 In the Components list, select MQSeries Server Receiver (alias: MqSeriesSrvRcvr).
- 6 Select the Parameters view tab under the Components list.
- 7 Set the following parameters as follows:

Parameter Name	Alias	Value
Receiver Connection Subsystem	ReceiverConnectionSubsystem	Subsystem name created in "To configure named subsystem" on page 63. For example, CIFMQConnSubsys or CIFMQDataSubsys.
Receiver Method Name	ReceiverMethodName	ReceiveDispatch or ReceiveDispatchSend.
Default Tasks	DfltTasks	1 or number of tasks desired.

- 8 Restart the Siebel Server and make sure the MQSeries Server 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. For complete instructions, see the procedure under ["Inbound Server Administration"](#) on page 62.

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.

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

6

Siebel Master Data Applications Messages

Siebel Master Data Applications provide application services interfaces to interact and manage the customer information stored in the Customer Master.

The application service interfaces supported with Siebel Master Data 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.
- **Manage Cross Reference**—Request and response message allowing querying of Contact, Account, and Household cross references.
- **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 lists each of the messages from Siebel Master Data Applications specifications.

Table 30. Siebel Universal Customer Master Application Messages

Integration Object Name	Business Processes/Operations
UCMAccountExternalID	LookUpAccountExternalID
UCMAccountExternalIDRs	LookUpAccountExternalIDResponse
UCMContactExternalID	LookUpContactExternalID
UCMContactExternalIDRs	LookUpContactExternalIDResponse
UCMHouseholdExternalID	LookUpHouseholdExternalID
UCMHouseholdExternalIDRs	LookUpHouseholdExternalIDResponse

Table 30. Siebel Universal Customer Master Application Messages

Integration Object Name	Business Processes/Operations
CIFAccount	LookUpAccount
CIFAccountRs	LookUpAccountResponse
CIFAccountActivity	LookUpAccountActivity
CIFAccountActivityRs	LookUpAccountActivityResponse
CIFAccountActivityByNameRs	LookUpAccountActivityByNameResponse
CIFAccountActivityDelete	DeleteAccountActivity
CIFAccountActivityDeleteRs	DeleteAccountActivityResponse
CIFAccountActivityInsert	InsertAccountActivity
CIFAccountActivityInsertRs	InsertAccountActivityResponse
CIFAccountActivityUpdate	UpdateAccountActivity
CIFAccountActivityUpdateRs	UpdateAccountActivityResponse
CIFAccountAddress	LookUpAccountAddress
CIFAccountAddressRs	LookUpAccountAddressResponse
CIFAccountAddressDelete	DeleteAccountAddress
CIFAccountAddressDeleteRs	DeleteAccountAddressResponse
CIFAccountAddressInsert	InsertAccountAddress
CIFAccountAddressInsertRs	InsertAccountAddressResponse
CIFAccountAddressUpdate	UpdateAccountAddress
CIFAccountAddressUpdateRs	UpdateAccountAddressResponse
CIFAccountDelete	DeleteAccount
CIFAccountDeleteRs	DeleteAccountResponse
CIFAccountInsert	InsertAccount
CIFAccountInsertRs	InsertAccountResponse
CIFAccountProduct	LookUpAccountProduct
CIFAccountProductRs	LookUpAccountProductResponse
CIFAccountProductDelete	DeleteAccountProduct
CIFAccountProductDeleteRs	DeleteAccountProductResponse
CIFAccountProductInsert	InsertAccountProduct
CIFAccountProductInsertRs	InsertAccountProductResponse
CIFAccountProductUpdate	UpdateAccountProduct
CIFAccountProductUpdateRs	UpdateAccountProductResponse

Table 30. Siebel Universal Customer Master Application Messages

Integration Object Name	Business Processes/Operations
CIFAccountProfile	LookUpAccountProfile
CIFAccountProfileRs	LookUpAccountProfileResponse
CIFAccountProfileByName	LookUpAccountProfileByName
CIFAccountProfileByNameRs	LookUpAccountProfileByNameResponse
CIFAccountProfileDelete	DeleteAccountProfile
CIFAccountProfileDeleteRs	DeleteAccountProfileResponse
CIFAccountProfileInsert	InsertAccountProfile
CIFAccountProfileInsertRs	InsertAccountProfileResponse
CIFAccountProfileUpdate	UpdateAccountProfile
CIFAccountProfileUpdateRs	UpdateAccountProfileResponse
CIFAccountRelationship	LookUpAccountRelationship
CIFAccountRelationshipRs	LookUpAccountRelationshipResponse
CIFAccountRelationshipDelete	DeleteAccountRelationship
CIFAccountRelationshipDeleteRs	DeleteAccountRelationshipResponse
CIFAccountRelationshipInsert	InsertAccountRelationship
CIFAccountRelationshipInsertRs	InsertAccountRelationshipResponse
CIFAccountRelationshipUpdate	UpdateAccountRelationship
CIFAccountRelationshipUpdateRs	UpdateAccountRelationshipResponse
CIFAccountUpdate	UpdateAccount
CIFAccountUpdateRs	UpdateAccountResponse
CIFContact	LookUpContact
CIFContactRs	LookUpContactResponse
CIFContactActivity	LookUpContactActivity
CIFContactActivityRs	LookUpContactActivityResponse
CIFContactActivityByName	LookUpContactActivityByName
CIFContactActivityByNameRs	LookUpContactActivityByNameResponse
CIFContactActivityDelete	DeleteContactActivity
CIFContactActivityDeleteRs	DeleteContactActivityResponse
CIFContactActivityInsert	InsertContactActivity
CIFContactActivityInsertRs	InsertContactActivityResponse
CIFContactActivityUpdate	UpdateContactActivity

Table 30. Siebel Universal Customer Master Application Messages

Integration Object Name	Business Processes/Operations
CIFContactActivityUpdateRs	UpdateContactActivityResponse
CIFContactAddress	LookUpContactAddress
CIFContactAddressRs	LookUpContactAddressResponse
CIFContactAddressByName	LookUpContactAddressByName
CIFContactAddressByNameRs	LookUpContactAddressByNameResponse
CIFContactAddressDelete	DeleteContactAddress
CIFContactAddressDeleteRs	DeleteContactAddressResponse
CIFContactAddressInsert	InsertContactAddress
CIFContactAddressInsertRs	InsertContactAddressResponse
CIFContactAddressUpdate	UpdateContactAddress
CIFContactAddressUpdateRs	UpdateContactAddressResponse
CIFContactByName	LookUpContactByName
CIFContactByNameRs	LookUpContactByNameResponse
CIFContactDelete	DeleteContact
CIFContactDeleteRs	DeleteContactResponse
CIFContactExternalId	LookUpContactExternalId
CIFContactExternalIdRs	LookUpContactExternalIdResponse
CIFContactInsert	InsertContact
CIFContactInsertRs	InsertContactResponse
CIFContactProduct	LookUpContactProduct
CIFContactProductRs	LookUpContactProductResponse
CIFContactProductByName	LookUpContactProductByName
CIFContactProductByNameRs	LookUpContactProductByNameResponse
CIFContactProductDelete	DeleteContactProduct
CIFContactProductDeleteRs	DeleteContactProductResponse
CIFContactProductInsert	InsertContactProduct
CIFContactProductInsertRs	InsertContactProductResponse
CIFContactProductUpdate	UpdateContactProduct
CIFContactProductUpdateRs	UpdateContactProductResponse
CIFContactProfile	LookUpContactProfile
CIFContactProfileRs	LookUpContactProfileResponse

Table 30. Siebel Universal Customer Master Application Messages

Integration Object Name	Business Processes/Operations
CIFContactProfileByName	LookUpContactProfileByName
CIFContactProfileByNameRs	LookUpContactProfileByNameResponse
CIFContactProfileDelete	DeleteContactProfile
CIFContactProfileDeleteRs	DeleteContactProfileResponse
CIFContactProfileInsert	InsertContactProfile
CIFContactProfileInsertRs	InsertContactProfileResponse
CIFContactProfileUpdate	UpdateContactProfile
CIFContactProfileUpdateRs	UpdateContactProfileResponse
CIFContactRelationship	LookUpContactRelationship
CIFContactRelationshipRs	LookUpContactRelationshipResponse
CIFContactRelationshipByName	LookUpContactRelationshipByName
CIFContactRelationshipByNameRs	LookUpContactRelationshipByNameResponse
CIFContactRelationshipDelete	DeleteContactRelationship
CIFContactRelationshipDeleteRs	DeleteContactRelationshipResponse
CIFContactRelationshipInsert	InsertContactRelationship
CIFContactRelationshipInsertRs	InsertContactRelationshipResponse
CIFContactRelationshipUpdate	UpdateContactRelationship
CIFContactRelationshipUpdateRs	UpdateContactRelationshipResponse
CIFContactUpdate	UpdateContact
CIFContactUpdateRs	UpdateContactResponse
CIFFINCORPAccount	LookUpFINCORPAccount
CIFFINCORPAccountRs	LookUpFINCORPAccountResponse
CIFFINCORPAccountByName	LookUpFINCORPAccountByName
CIFFINCORPAccountByNameRs	LookUpFINCORPAccountByNameResponse
CIFHousehold	LookUpHousehold
CIFHouseholdRs	LookUpHouseholdResponse
CIFHouseholdByName	LookUpHouseholdByName
CIFHouseholdByNameRs	LookUpHouseholdByNameResponse
CIFHouseholdDelete	DeleteHousehold
CIFHouseholdDeleteRs	DeleteHouseholdResponse
CIFHouseholdInsert	InsertHousehold

Table 30. Siebel Universal Customer Master Application Messages

Integration Object Name	Business Processes/Operations
CIFHouseholdInsertRs	InsertHouseholdResponse
CIFHouseholdUpdate	UpdateHousehold
CIFHouseholdUpdateRs	UpdateHouseholdResponse
CIFPersonalAddress	LookUpPersonalAddress
CIFPersonalAddressRs	LookUpPersonalAddressResponse
CIFPersonalAddressByState	LookUpPersonalAddressByState
CIFPersonalAddressByStateRs	LookUpPersonalAddressByStateResponse

About CRMML Envelope and Header Information

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

<! 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 UCM 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, there 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		

Table 31. Header Elements

Element	Structure	Attribute	Description	Default	Configuration
SystemInfo	In FromSystem		SystemInfo aggregate		
		Type	Describes the type of SystemID such as ID or URL	ID	UCM Envelope Integration Object->SystemInfo Integration Component->type Integration Component Field, XML Literal Value
		SystemId	Identifier of the system	123-456-789	UCM Envelope Integration Object->SystemInfo Integration Component->SystemId Integration Component Field, XML Literal Value
		SystemName	Name of the system	SiebelCIF System	UCM Envelope Integration Object->SystemInfo Integration Component->SystemName Integration Component Field, XML Literal Value
		SystemRole	Role the system has	Service Provider	UCM Envelope Integration Object->SystemInfo Integration Component->SystemRole Integration Component Field, XML Literal Value
ToSystem	In CRMHeader		ToSystem aggregate		

Table 31. Header Elements

Element	Structure	Attribute	Description	Default	Configuration
SystemInfo	In ToSystem		SystemInfo aggregate		
		Type	Describes the type of SystemID such as ID or URL.	ID	UCM 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
Message Descriptor	In CRMHeader	MessageId	Identifier of the message		System Generated
		Transaction Scope	Currently, the only value supported for transaction scope is any		UCM Envelope Integration Object->MessageDescriptor Integration Component->MessageDescriptor_TrasactionScope 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		UCM Envelope Integration Object->MessageDescriptor Integration Component->MessageDescriptor_ToLive Integration Component Field, XML Literal Value

Universally Unique Identifier

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 Master Data 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

UCM 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 14](#) describes the structure of this message.

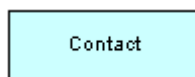


Figure 14. UCM Contact Profile Structure

Table 32. Contact Profile Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfile	Contact	LookUpContactProfile	1:PartyUID2: Id
CIFContactProfileRs	Contact	LookUpContactProfileResponse	1:PartyUID2: Id

Table 33. Contact Profile Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfileByName	Contact	LookUpContactProfileByName	1:PartyUID2: Id
CIFContactProfileByNameRs	Contact	LookUpContactProfileByNameResponse	1:PartyUID2: Id

Table 34. Contact Profile Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfileUpdate	Contact	UpdateContactProfile	1:PartyUID2: Id
CIFContactProfileUpdateRs	Contact	UpdateContactProfileResponse	1:PartyUID2: Id

Table 35. Contact Profile Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfileInsert	Contact	InsertContactProfile	1:PartyUID2: Id
CIFContactProfileUpdateRs	Contact	InsertContactProfileResponse	1:PartyUID2: Id

Table 36. Contact Profile Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfileDelete	Contact	DeleteContactProfile	1:PartyUID2: Id
CIFContactProfileDeleteRs	Contact	DeleteContactProfileResponse	1:PartyUID2: 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_TY PE
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	✓	
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	✓	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	✓	
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		
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		

Table 37. Contact Profile Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
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

UCM 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, InsertContactAdresse, and DeleteContactAddress are message pairs for request and response operations. [Figure 15](#) describes the structure of this message.

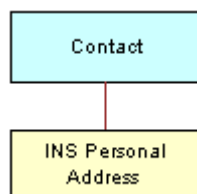


Figure 15. UCM Contact Address Structure

Table 38. Contact Address Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddress	Contact	LookUpContactAddress	1:PartyUID2: Id
CIFContactAddressRs	Contact	LookUpContactAddressResponse	1:PartyUID2: Id

Table 39. Contact Address Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddressByName	Contact	LookUpContactAddressByName	1:PartyUID2: Id
CIFContactAddressByNameRs	Contact	LookUpContactAddressByName Response	1:PartyUID2: Id

Table 40. Contact Address Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddressUpdate	Contact	UpdateContactAddress	1:PartyUID2: Id
CIFContactAddressUpdateRs	Contact	UpdateContactAddressResponse	1:PartyUID2: Id

Table 41. Contact Address Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddressInsert	Contact	InsertContactAddress	1:PartyUID2: Id
CIFContactAddressInsertRs	Contact	InsertContactAddressResponse	1:PartyUID2: Id

Table 42. Contact Address Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddressDelete	Contact	DeleteContactAddress	1:PartyUID2: Id
CIFContactAddressDeleteRs	Contact	DeleteContactAddressResponse	1:PartyUID2: 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	✓	
Id	Id	DTYPE_ID	15	✓	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	✓	
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		

Table 43. Contact Address Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Status	Status	DTYPE_TEXT	30		Bound by CONTACT_STAT US
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	✓	
INS Personal Country	INSPersonalCountry	DTYPE_TEXT	30		
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		
INS Personal Street Address	INSPersonalStreetAddress	DTYPE_TEXT	200	✓	
INS Personal Street Address 2	INSPersonalStreetAddress2	DTYPE_TEXT	100		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
operation	operation	DTYPE_TEXT	30		

Contact Activity

UCM 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 16 describes the structure of this message.

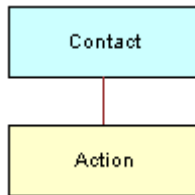


Figure 16. UCM Contact Activity Structure

Table 44. Contact Activity Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactActivity	Contact	LookUpContactActivity	1:PartyUID2: Id
CIFContactActivityRs	Contact	LookUpContactActivityResponse	1:PartyUID2: 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:PartyUID2: Id
CIFContactActivityUpdateRs	Contact	UpdateContactActivityResponse	1:PartyUID2: Id

Table 47. Contact Activity Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactActivityInsert	Contact	InsertContactActivity	1:PartyUID2: Id
CIFContactActivityInsertRs	Contact	InsertContactActivityResponse	1:PartyUID2: Id

Table 48. Contact Activity Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactActivityDelete	Contact	DeleteContactActivity	1:PartyUID2: Id
CIFContactActivityDeleteRs	Contact	DeleteContactActivityResponse	1:PartyUID2: 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	✓	
Home Phone #	HomePhone	DTYPE_TEXT	40		
Id	Id	DTYPE_ID	15	✓	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		

Table 49. Contact Activity Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Last Name	LastName	DTYPE_TEXT	50	✓	
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 UIId	PartyUIId	DTYPE_TEXT	100		
Person UIId	PersonUIId	DTYPE_TEXT	100		
Preferred Communications	PreferredCommunications	DTYPE_TEXT	30		Bound by OFFER_MEDIA
Status	Status	DTYPE_TEXT	30		Bound by CONTACT_STAT US
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		
Account Name	AccountName	DTYPE_TEXT	100		Used for picking a related entity.
Audience	Audience	DTYPE_TEXT	30		Bound by ACTIVITY_AUD IENCE_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		

Table 49. Contact Activity Integration Component Fields

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

Contact Product

UCM 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 17 describes the structure of this message

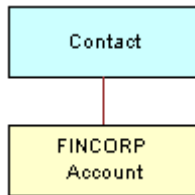


Figure 17. UCM Contact Product Structure

Table 50. Contact Product Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProduct	Contact	LookUpContactProduct	1:PartyUID2: Id
CIFContactProductRs	Contact	LookUpContactProductResponse	1:PartyUID2: Id

Table 51. Contact Product Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProductByName	Contact	LookUpContactProductByName	1:PartyUID2: Id
CIFContactProductByNameR	Contact	LookUpContactProductByNameResponse	1:PartyUID2: Id

Table 52. Contact Product Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProductUpdate	Contact	UpdateContactProduct	1:PartyUID2: Id
CIFContactProductUpdateRs	Contact	UpdateContactProductResponse	1:PartyUID2: Id

Table 53. Contact Product Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProductInsert	Contact	InsertContactProduct	1:PartyUID2: Id
CIFContactProductInsertRs	Contact	InsertContactProductResponse	1:PartyUID2: Id

Table 54. Contact Product Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProductDelete	Contact	DeleteContactProduct	1:PartyUID2: Id
CIFContactProductDeleteRs	Contact	DeleteContactProductResponse	1:PartyUID2: 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	✓	
Home Phone #	HomePhone	DTYPE_TEXT	40		
Id	Id	DTYPE_ID	15	✓	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		

Table 55. Contact Product Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Last Name	LastName	DTYPE_TEXT	50	✓	
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 UIId	PartyUIId	DTYPE_TEXT	100		
Person UIId	PersonUIId	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	✓	
Account Status	AccountStatus	DTYPE_TEXT	30		Bound by FINS_CRPST_L FLN_IPFS_PLC _MLOV
Account Type	AccountType	DTYPE_TEXT	30		Bound by FINCORP_PRO D_ADMIN_CLA SS_MLOV
Acct Gen - Branch Id	AcctGen-BranchId	DTYPE_ID	15		

Table 55. Contact Product Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
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	✓	Used for picking a related entity
Organization Id	OrganizationId	DTYPE_ID	15		
Product Id	ProductId	DTYPE_ID	15	✓	
Relationship Limit	RelationshipLimit	DTYPE_CURRENCY	22		
Start Date	StartDate	DTYPE_DATE	7		
operation	operation	DTYPE_TEXT	30		

Contact Relationship

UCM 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 18](#) describes the structure of this message

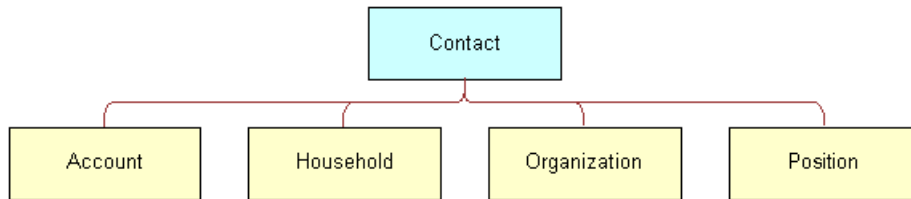


Figure 18. UCM Contact Relationship Structure

Table 56. Contact Relationship Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationship	Contact	LookUpContactRelationship	1:PartyUID2: Id
CIFContactRelationshipRs	Contact	LookUpContactRelationshipResponse	1:PartyUID2: Id

Table 57. Contact Relationship Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationshipByName	Contact	LookUpContactRelationshipByName	1:PartyUID2: Id
CIFContactRelationshipByNameRs	Contact	LookUpContactRelationshipByNameResponse	1:PartyUID2: Id

Table 58. Contact Relationship Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationshipUpdate	Contact	UpdateContactRelationship	1:PartyUID2:Id
CIFContactRelationshipUpdateRs	Contact	UpdateContactRelationshipResponse	1:PartyUID2:Id

Table 59. Contact Relationship Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationshipInsert	Contact	InsertContactRelationship	1:PartyUID2:Id
CIFContactRelationshipInsertRs	Contact	InsertContactRelationshipResponse	1:PartyUID2:Id

Table 60. Contact Relationship Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationshipDelete	Contact	DeleteContactRelationship	1:PartyUID2:Id
CIFContactRelationshipDeleteRs	Contact	DeleteContactRelationshipResponse	1:PartyUID2: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		

Table 61. Contact Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Email Address	EmailAddress	DTYPE_TEXT	50		
Fax Phone #	FaxPhone	DTYPE_TEXT	40		
First Name	FirstName	DTYPE_TEXT	50	✓	
Home Phone #	HomePhone	DTYPE_TEXT	40		
Id	Id	DTYPE_ID	15	✓	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	✓	
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		
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	ManagerPrimaryOrganization Id	DTYPE_ID	15		
Manager UIId	ManagerUIId	DTYPE_TEXT	100		
Middle Name	MiddleName	DTYPE_TEXT	50		
Party UIId	PartyUIId	DTYPE_TEXT	100		

Table 61. Contact Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Person UId	PersonUId	DTYPE_TEXT	100		
Preferred Communications	PreferredCommunications	DTYPE_TEXT	30		Bound by OFFER_MEDIA
Status	Status	DTYPE_TEXT	30		Bound by CONTACT_STAT US
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	✓	
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	30		Bound by ACCOUNT_STA TUS
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		

Table 61. Contact Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Household Status	HouseholdStatus	DTYPE_TEXT	30		Bound by HOUSEHOLD_S TATUS
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		
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

UCM 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 19 describes the structure of this message.

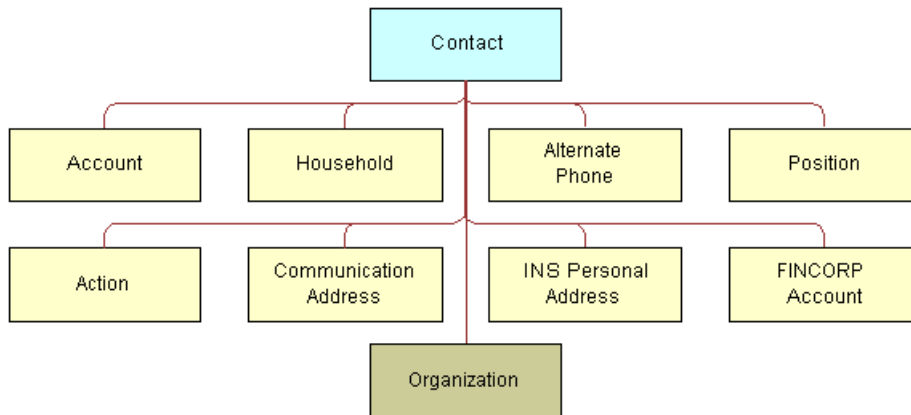


Figure 19. UCM Contact Structure

Table 62. Contact Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContact	Contact	LookUpContact	1:PartyUID2: Id
CIFContactRs	Contact	LookUpContactResponse	1:PartyUID2: Id

Table 63. Contact Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactByName	Contact	LookUpContactByName	1:PartyUID2: Id
CIFContactByNameRs	Contact	LookUpContactByNameResponse	1:PartyUID2: Id

Table 64. Contact Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactUpdate	Contact	UpdateContact	1:PartyUIId2: Id
CIFContactUpdateRs	Contact	UpdateContactResponse	1:PartyUIId2: Id

Table 65. Contact Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactInsert	Contact	InsertContact	1:PartyUIId2: Id
CIFContactInsertRs	Contact	InsertContactResponse	1:PartyUIId2: Id

Table 66. Contact Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactDelete	Contact	DeleteContact	1:PartyUIId2: Id
CIFContactDeleteRs	Contact	DeleteContactResponse	1:PartyUIId2: 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		

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Customer Value	CustomerValue	DTYPE_TEXT	30		Bound by FIN_CON_VALU E_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	✓	
Home Phone #	HomePhone	DTYPE_TEXT	40		
Id	Id	DTYPE_ID	15	✓	This tag is required for Id cross-referencing. Tag stores incoming external Id.
Income	Income	DTYPE_CURRENCY	22		
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	✓	
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		

Table 67. Contact 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	ManagerPrimaryOrganization Id	DTYPE_ID	15		
Manager UIId	ManagerUIId	DTYPE_TEXT	100		
Marital Status	MaritalStatus	DTYPE_TEXT	30		Bound by MARITAL_STAT US
Middle Name	MiddleName	DTYPE_TEXT	50		
Mother Maiden Name	MotherMaidenName	DTYPE_TEXT	50		
Party UIId	PartyUIId	DTYPE_TEXT	100		
Person UIId	PersonUIId	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_STAT US
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		

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
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
Account					
Account	Account	DTYPE_TEXT	100	✓	
Account Location	AccountLocation	DTYPE_TEXT	50		
Account Organization	AccountOrganization	DTYPE_TEXT	100		
Account Party UIId	AccountPartyUIId	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	✓	
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	✓	
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	✓	
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	HouseholdPartyUIId	DTYPE_TEXT	100		
Household Status	HouseholdStatus	DTYPE_TEXT	30		Bound by HOUSEHOLD_STATUS
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
operation	operation	DTYPE_TEXT	30		
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		

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
INS Personal City	INSPersonalCity	DTYPE_TEXT	50	✓	
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_ABBREV
INS Personal Street Address	INSPersonalStreetAddress	DTYPE_TEXT	200	✓	
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		
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		

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
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
Description	Description	DTYPE_TEXT	100		
Done	Done	DTYPE_UTC DATETIME	7		
Done Flag	DoneFlag	DTYPE_TEXT	1		
Due	Due	DTYPE_UTC DATETIME	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_UTC DATETIME	7		
Planned	Planned	DTYPE_UTC DATETIME	7		
Started	Started	DTYPE_UTC DATETIME	7		
Status	Status	DTYPE_TEXT	30		Bound by EVENT_STATUS
Type	Type	DTYPE_TEXT	30	✓	Bound by TODO_TYPE
operation	operation	DTYPE_TEXT	30		System field

Table 67. Contact 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		
Account Holder	AccountHolder	DTYPE_TEXT	100		Used for picking a related entity.
Account Name	AccountName	DTYPE_TEXT	100		
Account Number	AccountNumber3	DTYPE_TEXT	100	✓	
Account Status	AccountStatus2	DTYPE_TEXT	30		Bound by FINS_CRPST_L FLN_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	✓	Used for picking a related entity
Organization Id	OrganizationId2	DTYPE_ID	15		
Product Id	ProductId	DTYPE_ID	15	✓	
Relationship Limit	RelationshipLimit	DTYPE_CURRENCY	22		

Table 67. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Start Date	StartDate	DTYPE_DATE ETIME	7		
operation	operation	DTYPE_TEXT	30		System field

Household

UCM Household messages are used in insert, query, update, and delete operations to maintain Household information. LookUpHousehold and LookUpHouseholdByName are used for query operations. UpdateHousehold, InsertHousehold, and DeleteHousehold are message pairs for request and response operations. Figure 20 describes the structure of this message.

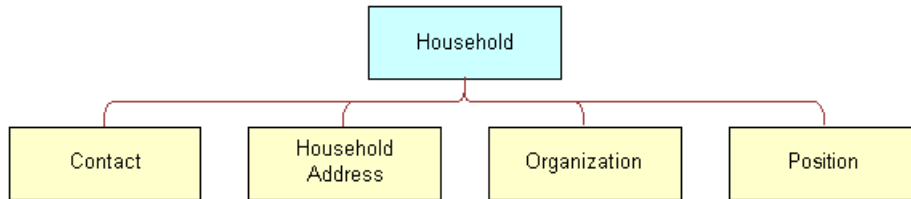


Figure 20. UCM Household Structure

Table 68. Household Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHousehold	Household	LookUpHousehold	1:PartyUID2: Id
CIFHouseholdRs	Household	LookUpHouseholdResponse	1:PartyUID2: Id

Table 69. Household Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHouseholdByName	Household	LookUpHouseholdByName	1:PartyUID2: Id
CIFHouseholdByNameRs	Household	LookUpHouseholdByNameResponse	1:PartyUID2: Id

Table 70. Household Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHouseholdUpdate	Household	UpdateHousehold	1:PartyUID2: Id
CIFHouseholdUpdateRs	Household	UpdateHouseholdResponse	1:PartyUID2: Id

Table 71. Household Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHouseholdInsert	Household	InsertHousehold	1:PartyUID2: Id
CIFHouseholdInsertRs	Household	InsertHouseholdResponse	1:PartyUID2: Id

Table 72. Household Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHouseholdDelete	Household	DeleteHousehold	1:PartyUID2: Id
CIFHouseholdDeleteRs	Household	DeleteHouseholdResponse	1:PartyUID2: 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		
Household Number	HouseholdNumber	DTYPE_TEXT	100		
Household Size	HouseholdSize	DTYPE_INTEGER	22		
Household Wealth	HouseholdWealth	DTYPE_CURRENCY	22		

Table 73. Household Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
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		
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	✓	
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		

Table 73. Household Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Last Name	LastName	DTYPE_TEXT	50	✓	
M/M	MM	DTYPE_TEXT	15		Bound by MR_MS
Middle Name	MiddleName	DTYPE_TEXT	50		
Person UId	PersonUIId	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_HOUSEHOLD
SSN	SSN	DTYPE_TEXT	11		
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	✓	
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_ABBREV
Street Address	StreetAddress	DTYPE_TEXT	200	✓	

Table 73. Household Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
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					
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

UCM Personal Address messages are used for requesting Personal Address information. They are a message pair for request and response operations. [Figure 21](#) describes the structure of this message.

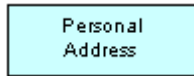


Figure 21. UCM Personal Address Structure

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	✓	
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		

Table 75. Personal Address Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
State	State	DTYPE_TEXT	10		Bound by STATE_ABBREV
Street Address	StreetAddress	DTYPE_TEXT	200	✓	
Street Address 2	StreetAddress2	DTYPE_TEXT	100		
Zip Code	ZipCode	DTYPE_TEXT	30		
operation	operation	DTYPE_TEXT	30		System field

FINCORP Account

UCM FINCORP Account messages are used for requesting FINCORP Account information. They are a message pair for request and response operations. Figure 22 describes the structure of this message.

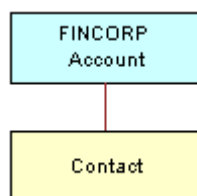


Figure 22. UCM FINCORP Account Structure

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

Table 77. FINCORP Account 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	AccountNumber	DTYPE_TEXT	100	✓	
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	✓	Used for picking a related entity.
Organization Id	OrganizationId	DTYPE_ID	15		
Parent ABA Number	ParentABANumber	DTYPE_TEXT	50		

Table 77. FINCORP Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
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	✓	
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	✓	
Acct Gen - Authorized Users - Last Name	AcctGen-AuthorizedUsers-LastName	DTYPE_TEXT	50	✓	
Acct Gen - Authorized Users - Party UIId	AcctGen-AuthorizedUsers-PartyUIId	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

UCM Account Profile messages are used in insert, query, update, and delete operations to maintain Account Profile Information. LookUpAccount Profile and LookUpProfileByName are used for query operations. UpdateAccountProfile, InsertAccountProfile, and DeleteAccountProfile are message pairs for request and response operations. [Figure 23](#) describes the structure of this message.

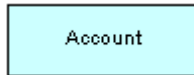


Figure 23. UCM Account Profile Structure

Table 78. Account Profile Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProfile	Account	LookUpAccountProfile	UK1:PartyUIIdUK2: Id
CIFAccountProfileRs	Account	LookUpAccountProfileResponse	UK1:PartyUIIdUK2: Id

Table 79. Account Profile Query by Name Integration Components

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

Table 80. Account Profile Update Integration Components

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

Table 81. Account Profile Insert Integration Components

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

Table 82. Account Profile Delete Integration Components

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

Table 83. Account Profile Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Account Region	AccountRegion	DTYPE_TEXT	30		Use for picking a related entity
Account Status	AccountStatus	DTYPE_TEXT	24		Bound by ACCOUNT_STATUS
Account Type Code	AccountTypeCode	DTYPE_TEXT	30	✓	Bound by CUT_ACCOUNT_TYPE
Account Value	AccountValue	DTYPE_TEXT	30		Bound by FIN_CON_VALUE_TYPE
Alias	Alias	DTYPE_TEXT	50		
Annual Revenue	AnnualRevenue	DTYPE_TEXT	22		
Client Flag	ClientFlag	DTYPE_BOOL	1		
Competitor	Competitor	DTYPE_BOOL	1		
Credit Agency Number	CreditAgencyNumber	DTYPE_TEXT	30		

Table 83. Account Profile Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Cumulative Sales	CumulativeSales	DTYPE_NUMBER	22		
Currency Code	CurrencyCode	DTYPE_TEXT	15	✓	
Current Volume	CurrentVolume	DTYPE_CURRENCY	22		
Dont Call	DontCall	DTYPE_BOOL	1		
Drug Spent Currency Code	DrugSpentCurrencyCode	DTYPE_TEXT	30	✓	Use for picking a related entity
Employees	Employees	DTYPE_NUMBER	22		
Facility Flag	FacilityFlag	DTYPE_BOOL	1		
Federal Tax Number	FederalTaxNumber	DTYPE_TEXT	15		
Financing Method	FinancingMethod	DTYPE_TEXT	30		Bound by FINCORP_CLIENT_FINANCING
Gross Potential Value	GrossPotentialValue	DTYPE_CURRENCY	22		
Group Type Code	GroupTypeCode	DTYPE_TEXT	30	✓	Bound by GROUP_TYPE_CD
Home Page	HomePage	DTYPE_TEXT	100		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Internal Org Flag	InternalOrgFlag	DTYPE_BOOL	1	✓	
Investor Flag	InvestorFlag	DTYPE_BOOL	1		
Last Year Sales	LastYearSales	DTYPE_CURRENCY	22		
Legal Status	LegalStatus	DTYPE_TEXT	30		Bound by UT_LEGAL_STATUS
Location	Location	DTYPE_TEXT	50		
Main Fax Number	MainFaxNumber	DTYPE_PHONE	40		
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40		
Master Account Id	MasterAccountId	DTYPE_ID	15		
Name	Name	DTYPE_TEXT	100	✓	

Table 83. Account Profile Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Partner Flag	PartnerFlag	DTYPE_BOOL	1		
Party UID	PartyUID	DTYPE_TEXT	100	✓	
Privacy Code	PrivacyCode	DTYPE_TEXT	30	✓	Bound by PRIVACY_CODE
Reference Flag	ReferenceFlag	DTYPE_BOOL	1		
Referenciable	Referenciable	DTYPE_BOOL	1		
Revenue Growth	RevenueGrowth	DTYPE_NUMBER	22		
Service Provider Flag	ServiceProviderFlag	DTYPE_TEXT	1		
Status	Status	DTYPE_TEXT	30		Bound by FINS_COMPETITOR _STATUS_MLOV
Total # Meters	TotalMeters	DTYPE_NUMBER	22		
Total Potential Volume	TotalPotentialVolume	DTYPE_CURRENCY	22		
Type	Type	DTYPE_TEXT	30	✓	Bound by ACCOUNT_TYPE
Unionized	Unionized	DTYPE_BOOL	1		
YTD Net Contribution	YTDNetContribution	DTYPE_CURRENCY	22		
operation	operation	DTYPE_TEXT	30		
Id	Id	DTYPE_ID	15		Use for cross-referencing

Account Address

UCM 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 24](#) describes the structure of this message.

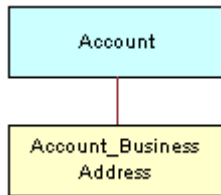


Figure 24. UCM Account Address Structure

Table 84. Account Address Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountAddress	Account	LookUpAccountAddress	UK1:PartyUIDUK2: Id
CIFAccountAddressRs	Account	LookUpAccountAddressResponse	UK1:PartyUIDUK2: Id

Table 85. Account Address Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountAddressByName	Account	LookUpAccountAddressByName	UK1:PartyUIDUK2: Id
CIFAccountAddressByNameRs	Account	LookUpAccountAddressByNameResponse	UK1:PartyUIDUK2: Id

Table 86. Account Address Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountAddressUpdate	Account	UpdateAccountAddress	UK1:PartyUIIdUK2:Id
CIFAccountAddressUpdateRs	Account	UpdateAccountAddressResponse	UK1:PartyUIIdUK2:Id

Table 87. Account Address Insert Integration Components

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

Table 88. Account Address Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountAddressDelete	Account	DeleteAccountAddress	UK1:PartyUIIdUK2:Id
CIFAccountAddressDeleteRs	Account	DeleteAccountAddressResponse	UK1:PartyUIIdUK2:Id

Table 89. Account Address Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Account					
Account Region	AccountRegion	DTYPE_TEXT	30		Use for picking a related entity
Account Status	AccountStatus	DTYPE_TEXT	24		Bound by ACCOUNT_STATUS

Table 89. Account Address Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Account Type Code	AccountTypeCode	DTYPE_TEXT	30	✓	Bound by CUT_ACCOUNT_TYPE
Account Value	AccountValue	DTYPE_TEXT	30		Bound by FIN_CON_VALUE_TYPE
Alias	Alias	DTYPE_TEXT	50		
Annual Revenue	AnnualRevenue	DTYPE_TEXT	22		
Client Flag	ClientFlag	DTYPE_BOOL	1		
Competitor	Competitor	DTYPE_BOOL	1		
Credit Agency Number	CreditAgencyNumber	DTYPE_TEXT	30		
Cumulative Sales	CumulativeSales	DTYPE_NUMBER	22		
Currency Code	CurrencyCode	DTYPE_TEXT	15	✓	
Current Volume	CurrentVolume	DTYPE_CURRENCY	22		
Dont Call	DontCall	DTYPE_BOOL	1		
Drug Spent Currency Code	DrugSpentCurrencyCode	DTYPE_TEXT	30	✓	Use for picking a related entity
Employees	Employees	DTYPE_NUMBER	22		
Facility Flag	FacilityFlag	DTYPE_BOOL	1		
Federal Tax Number	FederalTaxNumber	DTYPE_TEXT	15		
Financing Method	FinancingMethod	DTYPE_TEXT	30		Bound by FINCORP_CLIENT_FINANCING
Gross Potential Value	GrossPotentialValue	DTYPE_CURRENCY	22		
Group Type Code	GroupTypeCode	DTYPE_TEXT	30	✓	Bound by GROUP_TYPE_CD
Home Page	HomePage	DTYPE_TEXT	100		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Internal Org Flag	InternalOrgFlag	DTYPE_BOOL	1	✓	

Table 89. Account Address Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Investor Flag	InvestorFlag	DTYPE_BOOL	1		
Last Year Sales	LastYearSales	DTYPE_CURRENCY	22		
Legal Status	LegalStatus	DTYPE_TEXT	30		Bound by UT_LEGAL_STATUSES
Location	Location	DTYPE_TEXT	50		
Main Fax Number	MainFaxNumber	DTYPE_PHONE	40		
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40		
Master Account Id	MasterAccountId	DTYPE_ID	15		
Name	Name	DTYPE_TEXT	100	✓	
Partner Flag	PartnerFlag	DTYPE_BOOL	1		
Party UId	PartyUId	DTYPE_TEXT	100	✓	
Privacy Code	PrivacyCode	DTYPE_TEXT	30	✓	Bound by PRIVACY_CODE
Reference Flag	ReferenceFlag	DTYPE_BOOL	1		
Referenciable	Referenciable	DTYPE_BOOL	1		
Revenue Growth	RevenueGrowth	DTYPE_NUMBER	22		
Service Provider Flag	ServiceProviderFlag	DTYPE_TEXT	1		
Status	Status	DTYPE_TEXT	30		Bound by FINS_COMPETITOR_STATUS_MLOV
Total # Meters	TotalMeters	DTYPE_NUMBER	22		
Total Potential Volume	TotalPotentialVolume	DTYPE_CURRENCY	22		
Type	Type	DTYPE_TEXT	30	✓	Bound by ACCOUNT_TYPE
Unionized	Unionized	DTYPE_BOOL	1		
YTD Net Contribution	YTDNetContribution	DTYPE_CURRENCY	22		

Table 89. Account Address Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
operation	operation	DTYPE_TEXT	30		
Id	Id	DTYPE_ID	15		For cross-referencing
Account_Business Address					
Address Id	AddressId	DTYPE_ID	30	✓	
Address Integration Id	AddressIntegrationId	DTYPE_TEXT	30		
Bill Address Flag	BillAddressFlag	DTYPE_TEXT	1		
City	City	DTYPE_TEXT	50		
Country	Country	DTYPE_TEXT	50		Use for pick a related entity
Email Address	EmailAddress	DTYPE_TEXT	50		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Postal Code	PostalCode	DTYPE_TEXT	30		
Ship Address Flag	ShipAddressFlag	DTYPE_TEXT	1		
State	State	DTYPE_TEXT	10		Bound by STATE_ABBREV
Street Address	StreetAddress	DTYPE_TEXT	200		
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		

Account Activity

UCM 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 25 describes the structure of this message.

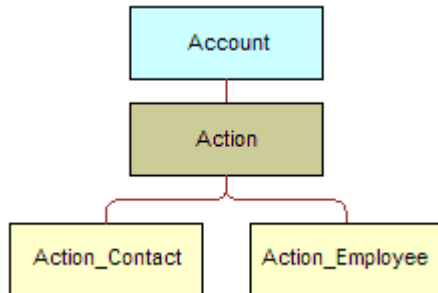


Figure 25. UCM Account Activity Structure

Table 90. Account Activity Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountActivity	Account	LookUpAccountActivity	UK1:PartyUIIdUK2:Id
CIFAccountActivityRs	Account	LookUpAccountActivityResponse	UK1:PartyUIIdUK2:Id

Table 91. Account Activity Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountActivityByName	Account	LookUpAccountActivityByNames	UK1:PartyUIIdUK2:Id
CIFAccountActivityByNameRs	Account	LookUpAccountActivityByNameResponse	UK1:PartyUIIdUK2:Id

Table 92. Account Activity Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountActivityUpdate	Account	UpdateAccountActivitys	UK1:PartyUIIdUK2: Id
CIFAccountActivityUpdateRs	Account	UpdateAccountActivityResponse	UK1:PartyUIIdUK2: Id

Table 93. Account Activity Insert Integration Components

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

Table 94. Account Activity Delete Integration Components

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

Table 95. Account Activity Integration Component Fields

Name	XML Tag	Type	Length	Req	Notes
Account					
Account Region	AccountRegion	DTYPE_TEXT	30		Use for picking a related entity
Account Status	AccountStatus	DTYPE_TEXT	24		Bound by ACCOUNT_STATUS

Table 95. Account Activity Integration Component Fields

Name	XML Tag	Type	Length	Req	Notes
Account Type Code	AccountTypeCode	DTYPE_TEXT	30	✓	Bound by CUT_ACCOUNT_TYPE
Account Value	AccountValue	DTYPE_TEXT	30		Bound by FIN_CON_VALUE_TYPE
Alias	Alias	DTYPE_TEXT	50		
Annual Revenue	AnnualRevenue	DTYPE_TEXT	22		
Client Flag	ClientFlag	DTYPE_BOOL	1		
Competitor	Competitor	DTYPE_BOOL	1		
Credit Agency Number	CreditAgencyNumber	DTYPE_TEXT	30		
Cumulative Sales	CumulativeSales	DTYPE_NUMBER	22		
Currency Code	CurrencyCode	DTYPE_TEXT	15	✓	
Current Volume	CurrentVolume	DTYPE_CURRENCY	22		
Dont Call	DontCall	DTYPE_BOOL	1		
Drug Spent Currency Code	DrugSpentCurrencyCode	DTYPE_TEXT	30	✓	Use for picking a related entity
Employees	Employees	DTYPE_NUMBER	22		
Facility Flag	FacilityFlag	DTYPE_BOOL	1		
Federal Tax Number	FederalTaxNumber	DTYPE_TEXT	15		
Financing Method	FinancingMethod	DTYPE_TEXT	30		Bound by FINCORP_CLIENT_FINANCING
Gross Potential Value	GrossPotentialValue	DTYPE_CURRENCY	22		
Group Type Code	GroupTypeCode	DTYPE_TEXT	30	✓	Bound by GROUP_TYPE_CD
Home Page	HomePage	DTYPE_TEXT	100		
Integration Id	IntegrationId	DTYPE_TEXT	30		

Table 95. Account Activity Integration Component Fields

Name	XML Tag	Type	Length	Req	Notes
Internal Org Flag	InternalOrgFlag	DTYPE_BOOL	1	✓	
Investor Flag	InvestorFlag	DTYPE_BOOL	1		
Last Year Sales	LastYearSales	DTYPE_CURRENCY	22		
Legal Status	LegalStatus	DTYPE_TEXT	30		Bound by UT_LEGAL_STATUSES
Location	Location	DTYPE_TEXT	50		
Main Fax Number	MainFaxNumber	DTYPE_PHONE	40		
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40		
Master Account Id	MasterAccountId	DTYPE_ID	15		
Name	Name	DTYPE_TEXT	100	✓	
Partner Flag	PartnerFlag	DTYPE_BOOL	1		
Party UId	PartyUId	DTYPE_TEXT	100	✓	
Privacy Code	PrivacyCode	DTYPE_TEXT	30	✓	Bound by PRIVACY_CODE
Reference Flag	ReferenceFlag	DTYPE_BOOL	1		
Referenciable	Referenciable	DTYPE_BOOL	1		
Revenue Growth	RevenueGrowth	DTYPE_NUMBER	22		
Service Provider Flag	ServiceProviderFlag	DTYPE_TEXT	1		
Status	Status	DTYPE_TEXT	30		Bound by FINS_COMPETITOR_STATUS_MLOV
Total # Meters	TotalMeters	DTYPE_NUMBER	22		
Total Potential Volume	TotalPotentialVolume	DTYPE_CURRENCY	22		
Type	Type	DTYPE_TEXT	30	✓	Bound by ACCOUNT_TYPE
Unionized	Unionized	DTYPE_BOOL	1		

Table 95. Account Activity Integration Component Fields

Name	XML Tag	Type	Length	Req	Notes
YTD Net Contribution	YTDNetContribution	DTYPE_CURRENCY	22		
operation	operation	DTYPE_TEXT	30		
Id	Id	DTYPE_ID	15		For cross-referencing
Action					
Alarm	Alarm	DTYPE_BOOL	1		
Assignment Excluded	AssignmentExcluded	DTYPE_BOOL	1		
Associated Cost	AssociatedCost	DTYPE_CURRENCY	22		
Billable Flag	BillableFlag	DTYPE_BOOL	1		
Call Id	CallId	DTYPE_TEXT	30		
Category	Category	DTYPE_TEXT	30	✓	Use for picking a related entity
Class	Class	DTYPE_TEXT	30		Bound by FS_ACTIVITY_CLASS
Comment	Comment	DTYPE_TEXT	"1,500"		
Description	Description	DTYPE_TEXT	100		
Display	Display	DTYPE_TEXT	30		
Done	Done	DTYPE_UTCDATETIME	7		
Done Flag	DoneFlag	DTYPE_BOOL	1		
Due	Due	DTYPE_UTCDATETIME	7		
Duration Minutes	DurationMinutes	DTYPE_INTEGER	22		Bound by APPOINTMENT_DURATION
Expense Related Flag	ExpenseRelatedFlag	DTYPE_BOOL	1		
Id	Id	DTYPE_TEXT	100	✓	
Integration Id	IntegrationId	DTYPE_TEXT	30		
No Sooner Than Date	NoSoonerThanDate	DTYPE_UTCDATETIME	7		

Table 95. Account Activity Integration Component Fields

Name	XML Tag	Type	Length	Req	Notes
Objective	Objective	DTYPE_TEXT	30		
Owned By	OwnedBy	DTYPE_TEXT	30	✓	Use for pick a related entity
Owned By Id	OwnedById	DTYPE_TEXT	30	✓	
Primary Owned By	PrimaryOwnedBy	DTYPE_TEXT	50		Use for pick a related entity
Primary Owner Id	PrimaryOwnerId	DTYPE_ID	30		
Priority	Priority	DTYPE_TEXT	30		Bound by ACTIVITY_PRIORITY
Private	Private	DTYPE_BOOL	1		
Repeating	Repeating	DTYPE_BOOL	1		
Repeating Expires	RepeatingExpires	DTYPE_DATETIME	7		
Status	Status	DTYPE_TEXT	30		Bound by EVENT_STATUS
Type	Type2	DTYPE_TEXT	30	✓	Bound by TODO_TYPE
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
Action_Contact					
Company Identifier	CompanyIdentifier	DTYPE_TEXT	15		
Contact First Name	ContactFirstName	DTYPE_TEXT	50	✓	
Contact Last Name	ContactLastName	DTYPE_TEXT	50	✓	
Employee Number	EmployeeNumber	DTYPE_TEXT	30		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Personal Contact	PersonalContact	DTYPE_BOOL	1		
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		

Table 95. Account Activity Integration Component Fields

Name	XML Tag	Type	Length	Req	Notes
Action_Employee					
Emp #	Emp	DTYPE_TEXT	30		
Employee First Name (MM)	EmployeeFirstNameMM	DTYPE_TEXT	50	✓	
Employee Job Title (MM)	EmployeeJobTitleMM	DTYPE_TEXT	75		
Employee Last Name (MM)	EmployeeLastNameMM	DTYPE_TEXT	50	✓	
Employee Login Name (MM)	EmployeeLoginNameMM	DTYPE_TEXT	50	✓	
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Private	Private	DTYPE_BOOL	1	✓	
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		

Account Product

UCM 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 26 describes the structure of this message.

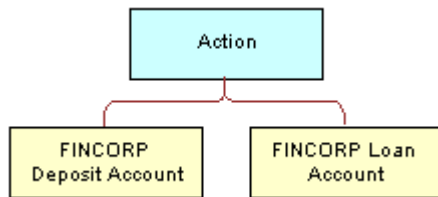


Figure 26. UCM Account Products Structure

Table 96. Account Product Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProduct	Account	LookUpAccountProduct	UK1:PartyUIIdUK2: Id
CIFAccountProductRs	Account	LookUpAccountProductResponse	UK1:PartyUIIdUK2: Id

Table 97. Account Product Query by Name Integration Components

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

Table 98. Account Product Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProductUpdate	Account	UpdateAccountProduct	UK1:PartyUIIdUK2:Id
CIFAccountProductUpdateRs	Account	UpdateAccountProductResponse	UK1:PartyUIIdUK2:Id

Table 99. Account Product Insert Integration Components

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

Table 100. Account Product Delete Integration Components

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

Table 101. Account Product Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Account					
Account Region	AccountRegion	DTYPE_TEXT	30		Use for picking a related entity
Account Status	AccountStatus	DTYPE_TEXT	24		Bound by ACCOUNT_STATUS

Table 101. Account Product Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Account Type Code	AccountTypeCode	DTYPE_TEXT	30	✓	Bound by CUT_ACCOUNT_TYPE
Account Value	AccountValue	DTYPE_TEXT	30		Bound by FIN_CON_VALUE_TYPE
Alias	Alias	DTYPE_TEXT	50		
Annual Revenue	AnnualRevenue	DTYPE_TEXT	22		
Client Flag	ClientFlag	DTYPE_BOOL	1		
Competitor	Competitor	DTYPE_BOOL	1		
Credit Agency Number	CreditAgencyNumber	DTYPE_TEXT	30		
Cumulative Sales	CumulativeSales	DTYPE_NUMBER	22		
Currency Code	CurrencyCode	DTYPE_TEXT	15	✓	
Current Volume	CurrentVolume	DTYPE_CURRENCY	22		
Dont Call	DontCall	DTYPE_BOOL	1		
Drug Spent Currency Code	DrugSpentCurrencyCode	DTYPE_TEXT	30	✓	Use for picking a related entity
Employees	Employees	DTYPE_NUMBER	22		
Facility Flag	FacilityFlag	DTYPE_BOOL	1		
Federal Tax Number	FederalTaxNumber	DTYPE_TEXT	15		
Financing Method	FinancingMethod	DTYPE_TEXT	30		Bound by FINCORP_CLIENT_FINANCING
Gross Potential Value	GrossPotentialValue	DTYPE_CURRENCY	22		
Group Type Code	GroupTypeCode	DTYPE_TEXT	30	✓	Bound by GROUP_TYPE_CD
Home Page	HomePage	DTYPE_TEXT	100		
Integration Id	IntegrationId	DTYPE_TEXT	30		

Table 101. Account Product Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Internal Org Flag	InternalOrgFlag	DTYPE_BOOL	1	✓	
Investor Flag	InvestorFlag	DTYPE_BOOL	1		
Last Year Sales	LastYearSales	DTYPE_CURRENCY	22		
Legal Status	LegalStatus	DTYPE_TEXT	30		Bound by UT_LEGAL_STATUS
Location	Location	DTYPE_TEXT	50		
Main Fax Number	MainFaxNumber	DTYPE_PHONE	40		
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40		
Master Account Id	MasterAccountId	DTYPE_ID	15		
Name	Name	DTYPE_TEXT	100		
Partner Flag	PartnerFlag	DTYPE_BOOL	1		
Party UId	PartyUId	DTYPE_TEXT	100	✓	
Privacy Code	PrivacyCode	DTYPE_TEXT	30	✓	Bound by PRIVACY_CODE
Reference Flag	ReferenceFlag	DTYPE_BOOL	1		
Referenciable	Referenciable	DTYPE_BOOL	1		
Revenue Growth	RevenueGrowth	DTYPE_NUMBER	22		
Service Provider Flag	ServiceProviderFlag	DTYPE_TEXT	1		
Status	Status	DTYPE_TEXT	30		Bound by FINS_COMPETITOR_STATUS_MLOV
Total # Meters	TotalMeters	DTYPE_NUMBER	22		
Total Potential Volume	TotalPotentialVolume	DTYPE_CURRENCY	22		
Type	Type	DTYPE_TEXT	30		Bound by ACCOUNT_TYPE
Unionized	Unionized	DTYPE_BOOL	1		

Table 101. Account Product Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
YTD Net Contribution	YTDNetContribution	DTYPE_CURRENCY	22		
operation	operation	DTYPE_TEXT	30		
Id	Id	DTYPE_ID	15		For cross-referencing
FINCORP Deposit Account					
Account Branch Number	AccountBranchNumber	DTYPE_TEXT	100		
Account Category	AccountCategory	DTYPE_TEXT	30		Bound by FINS_ASSET_PROD_CLASS_MLOV
Account Number	AccountNumber	DTYPE_TEXT	100	✓	
Account Status	AccountStatus	DTYPE_TEXT	30		Bound by FINS_CRPST_LFLN_IPFS_PLC_MLOV
Branch	Branch	DTYPE_TEXT	50		
Current Balance	CurrentBalance	DTYPE_CURRENCY	22		
Id	Id	DTYPE_TEXT	100		
Integration Id	Integration Id	DTYPE_TEXT	30		
Name	Name	DTYPE_TEXT	100	✓	Use for pick a related entity
Product Id	ProductId	DTYPE_ID	100	✓	
Type	Type	DTYPE_TEXT	30		Bound by FINS_OWNERSHIP_TYPE_MLOV
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
FINCORP Loan Account					
Account Branch Number	AccountBranchNumber	DTYPE_TEXT	30		

Table 101. Account Product Integration Component Fields

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

Account Relationship

UCM 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 27 describes the structure of this message.

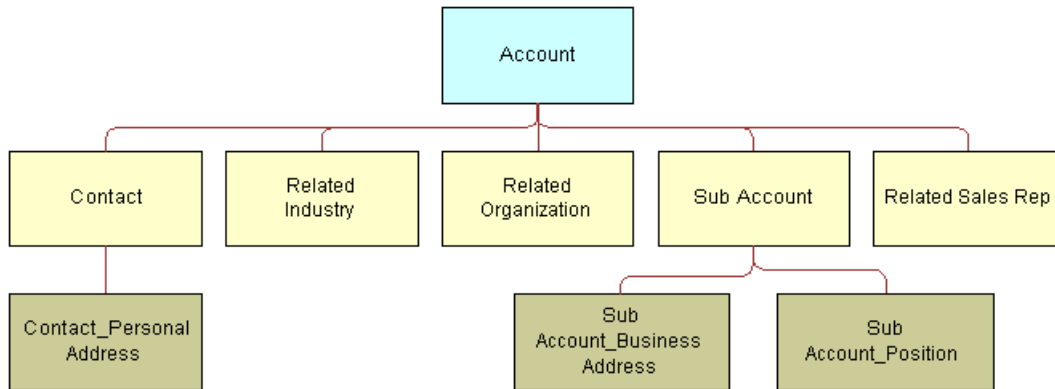


Figure 27. UCM Account Relationship Structure

Table 102. Account Relationship Query Integration Components

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

Table 103. Account Relationship Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountRelationshipByName	Account	LookUpAccountRelationshipByName	UK1:PartyUIId UK2: Id
CIFAccountRelationshipByNameRs	Account	LookUpAccountRelationshipByName Response	UK1:PartyUIId UK2: Id

Table 104. Account Relationship Update Integration Components

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

Table 105. Account Relationship Insert Integration Components

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

Table 106. Account Relationship Delete Integration Components

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

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Account					
Account Region	AccountRegion	DTYPE_TEXT	30		Use for picking a related entity
Account Status	AccountStatus	DTYPE_TEXT	24		Bound by ACCOUNT_STATUS

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Account Type Code	AccountTypeCode	DTYPE_TEXT	30	✓	Bound by CUT_ACCOUNT_TYPE
Account Value	AccountValue	DTYPE_TEXT	30		Bound by FIN_CON_VALUE_TYPE
Alias	Alias	DTYPE_TEXT	50		
Annual Revenue	AnnualRevenue	DTYPE_TEXT	22		
Client Flag	ClientFlag	DTYPE_BOOL	1		
Competitor	Competitor	DTYPE_BOOL	1		
Credit Agency Number	CreditAgencyNumber	DTYPE_TEXT	30		
Cumulative Sales	CumulativeSales	DTYPE_NUMBER	22		
Currency Code	CurrencyCode	DTYPE_TEXT	15	✓	
Current Volume	CurrentVolume	DTYPE_CURRENCY	22		
Dont Call	DontCall	DTYPE_BOOL	1		
Drug Spent Currency Code	DrugSpentCurrencyCode	DTYPE_TEXT	30	✓	Use for picking a related entity
Employees	Employees	DTYPE_NUMBER	22		
Facility Flag	FacilityFlag	DTYPE_BOOL	1		
Federal Tax Number	FederalTaxNumber	DTYPE_TEXT	15		
Financing Method	FinancingMethod	DTYPE_TEXT	30		Bound by FINCORP_CLIENT_FINANCING
Gross Potential Value	GrossPotentialValue	DTYPE_CURRENCY	22		
Group Type Code	GroupTypeCode	DTYPE_TEXT	30	✓	Bound by GROUP_TYPE_CD
Home Page	HomePage	DTYPE_TEXT	100		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Internal Org Flag	InternalOrgFlag	DTYPE_BOOL	1	✓	

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Investor Flag	InvestorFlag	DTYPE_BOOL	1		
Last Year Sales	LastYearSales	DTYPE_CURRENCY	22		
Legal Status	LegalStatus	DTYPE_TEXT	30		Bound by UT_LEGAL_STATUS
Location	Location	DTYPE_TEXT	50		
Main Fax Number	MainFaxNumber	DTYPE_PHONE	40		
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40		
Master Account Id	MasterAccountId	DTYPE_ID	15		
Name	Name	DTYPE_TEXT	100	✓	
Partner Flag	PartnerFlag	DTYPE_BOOL	1		
Party UID	PartyUid	DTYPE_TEXT	100	✓	
Privacy Code	PrivacyCode	DTYPE_TEXT	30	✓	Bound by PRIVACY_CODE
Reference Flag	ReferenceFlag	DTYPE_BOOL	1		
Referenciable	Referenciable	DTYPE_BOOL	1		
Revenue Growth	RevenueGrowth	DTYPE_NUMBER	22		
Service Provider Flag	ServiceProviderFlag	DTYPE_TEXT	1		
Status	Status	DTYPE_TEXT	30		Bound by FINS_COMPETITOR_STATUS_MLOV
Total # Meters	TotalMeters	DTYPE_NUMBER	22		
Total Potential Volume	TotalPotentialVolume	DTYPE_CURRENCY	22		
Type	Type	DTYPE_TEXT	30		Bound by ACCOUNT_TYPE
Unionized	Unionized	DTYPE_BOOL	1		
YTD Net Contribution	YTDNetContribution	DTYPE_CURRENCY	22		

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
operation	operation	DTYPE_TEXT	30		
Id	Id	DTYPE_ID	15		For cross-referencing
Contact					
Account Id	AccountId	DTYPE_ID	30		
Account Integration Id	AccountIntegrationId	DTYPE_TEXT	30		
Assistant	Assistant	DTYPE_TEXT	50		
Assistant Phone #	AssistantPhone	DTYPE_PHONE	40		
Cellular Phone #	CellularPhone	DTYPE_PHONE	40		
Fax Phone #	FaxPhone	DTYPE_PHONE	40		
First Name	FirstName	DTYPE_TEXT	50	✓	
Home Phone #	HomePhone	DTYPE_TEXT	40		
Job Title	JobTitle	DTYPE_TEXT	75		
Last Name	LastName	DTYPE_TEXT	50	✓	
M/M	MM	DTYPE_TEXT	15		
Middle Name	MiddleName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100	✓	
Person UId	PersonUId	DTYPE_ID	30		
Preferred Communication Method	PreferredCommunicationMethod	DTYPE_TEXT	20		Bound by COMM_METHOD
Privacy Code	PrivacyCode	DTYPE_TEXT	30	✓	Bound by PRIVACY_CODE
Row Id	RowId	DTYPE_ID	100		
Work Phone #	WorkPhone	DTYPE_PHONE	45		
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
Contact_Personal Address					
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Personal Address Id	PersonalAddressId	DTYPE_ID	100	✓	
Personal Address Integration Id	PersonalAddressIntegrationId	DTYPE_TEXT	30		
Personal City	PersonalCity	DTYPE_TEXT	50	✓	
Personal Country	PersonalCountry	DTYPE_TEXT	50		Bound by COUNTRY
Personal Postal Code	PersonalPostalCode	DTYPE_TEXT	30		
Personal State	PersonalState	DTYPE_TEXT	10		Bound by STATE_ABBREV
Personal Street Address	PersonalStreetAddress	DTYPE_TEXT	200	✓	
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
Related Industry					
Industry	Industry	DTYPE_TEXT	50	✓	
Industry Id	IndustryId	DTYPE_ID	30	✓	
Industry Integration Id	IndustryIntegrationId	DTYPE_TEXT	30		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
SIC Code	SICCode	DTYPE_TEXT	30	✓	
Related Organization					
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Organization	Organization	DTYPE_TEXT	100	✓	
Organization Id	OrganizationId	DTYPE_ID	100	✓	
Organization Integration Id	OrganizationIntegrationId	DTYPE_TEXT	30		
Related Sales Reps					
Division	Division	DTYPE_TEXT	255	✓	Use to pick a related entity
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Login	Login	DTYPE_ID	15		

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Position	Position	DTYPE_TEXT	50		
Position Id	PositionId	DTYPE_ID	30		
Position Integration Id	PositionIntegrationId	DTYPE_TEXT	30		
Sub Account					
Account Status	AccountStatus2	DTYPE_TEXT	30		Bound by ACCOUNT_STATUS
Id	Id	DTYPE_ID	100	✓	
Location	Location	DTYPE_TEXT	50		
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40		
Name	Name	DTYPE_TEXT	100	✓	
Party UID	PartyUID	DTYPE_TEXT	30	✓	
Type	Type	DTYPE_TEXT	255	✓	Bound by ACCOUNT_TYPE
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
Party UID	PartyUID	DTYPE_TEXT	30		
Type	Type	DTYPE_TEXT	255		
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
Sub Account_Business Address					
Address Id	AddressId	DTYPE_ID	30	✓	
Address Name	AddressName	DTYPE_TEXT	100		
Integration Id	IntegrationId	DTYPE_TEXT	30		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Street Address	StreetAddress	DTYPE_TEXT	200	✓	
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
Sub Account_Position					
Division	Division	DTYPE_TEXT	255	✓	Use to pick the related entity

Table 107. Account Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Division Integration Id	DivisionIntegrationId	DTYPE_TEXT	30		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Position	Position	DTYPE_TEXT	50		Bound by POSITION_TYPE
Position Id	PositionId	DTYPE_ID	30		
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		

Account

UCM Account messages are used in insert, query, update, and delete operations to maintain Account Profile Information. LookUpAccount Profile and LookUpProfileByName are used for query operations. UpdateAccountProfile, InsertAccountProfile, and DeleteAccountProfile are message pairs for request and response operations. Figure 28 describes the structure of this message.

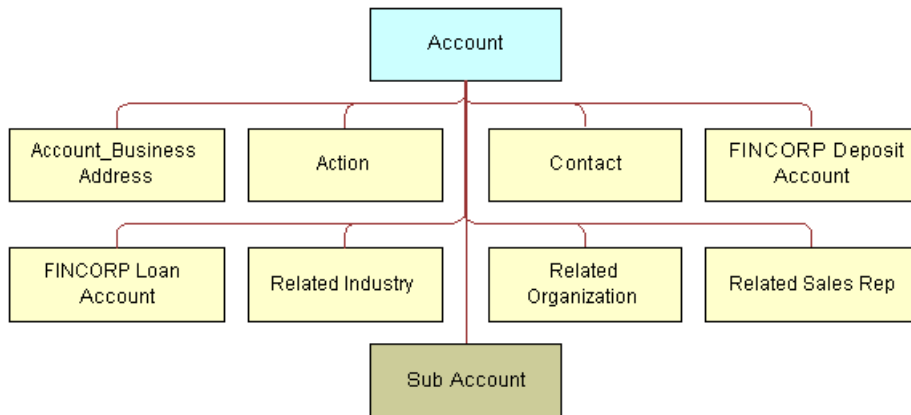


Figure 28. UCM Account Structure

Table 108. Account Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccount	Account	LookUpAccount	UK1:PartyUIIdUK2: Id
CIFAccountRs	Account	LookUpAccountResponse	UK1:PartyUIIdUK2: Id

Table 109. Account Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountUpdate	Account	UpdateAccount	UK1:PartyUIIdUK2: Id
CIFAccountUpdateRs	Account	UpdateAccountResponse	UK1:PartyUIIdUK2: Id

Table 110. Account Update Integration Components

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

Table 111. Account Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountInsert	Account	InsertAccount	UK1:PartyUIIdUK2: Id
CIFAccountInsertRs	Account	InsertAccountResponse	UK1:PartyUIIdUK2: Id

Table 112. Account Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountDelete	Account	DeleteAccount	UK1:PartyUIIdUK2: Id
CIFAccountDeleteRs	Account	DeleteAccountResponse	UK1:PartyUIIdUK2: Id

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Account					
Account Region	AccountRegion	DTYPE_TEXT	30		Use for picking a related entity
Account Status	AccountStatus	DTYPE_TEXT	24		Bound by ACCOUNT_STATUS
Account Type Code	AccountTypeCode	DTYPE_TEXT	30	✓	Bound by CUT_ACCOUNT_TYPE

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Account Value	AccountValue	DTYPE_TEXT	30		Bound by FIN_CON_VALUE_T YPE
Alias	Alias	DTYPE_TEXT	50		
Annual Revenue	AnnualRevenue	DTYPE_TEXT	22		
Client Flag	ClientFlag	DTYPE_BOOL	1		
Competitor	Competitor	DTYPE_BOOL	1		
Credit Agency Number	CreditAgencyNumber	DTYPE_TEXT	30		
Cumulative Sales	CumulativeSales	DTYPE_NUMBER	22		
Currency Code	CurrencyCode	DTYPE_TEXT	15	✓	
Current Volume	CurrentVolume	DTYPE_CURRENCY	22		
Dont Call	DontCall	DTYPE_BOOL	1		
Drug Spent Currency Code	DrugSpentCurrencyCode	DTYPE_TEXT	30	✓	Use for picking a related entity
Employees	Employees	DTYPE_NUMBER	22		
Facility Flag	FacilityFlag	DTYPE_BOOL	1		
Federal Tax Number	FederalTaxNumber	DTYPE_TEXT	15		
Financing Method	FinancingMethod	DTYPE_TEXT	30		Bound by FINCORP_CLIENT_ FINANCING
Gross Potential Value	GrossPotentialValue	DTYPE_CURRENCY	22		
Group Type Code	GroupTypeCode	DTYPE_TEXT	30	✓	Bound by GROUP_TYPE_CD
Home Page	HomePage	DTYPE_TEXT	100		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Internal Org Flag	InternalOrgFlag	DTYPE_BOOL	1	✓	
Investor Flag	InvestorFlag	DTYPE_BOOL	1		

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Last Year Sales	LastYearSales	DTYPE_CURRENCY	22		
Legal Status	LegalStatus	DTYPE_TEXT	30		Bound by UT_LEGAL_STATUS
Location	Location	DTYPE_TEXT	50		
Main Fax Number	MainFaxNumber	DTYPE_PHONE	40		
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40		
Master Account Id	MasterAccountId	DTYPE_ID	15		
Name	Name	DTYPE_TEXT	100	✓	
Partner Flag	PartnerFlag	DTYPE_BOOL	1		
Party UId	PartyUId	DTYPE_TEXT	100	✓	
Privacy Code	PrivacyCode	DTYPE_TEXT	30	✓	Bound by PRIVACY_CODE
Reference Flag	ReferenceFlag	DTYPE_BOOL	1		
Referenciable	Referenciable	DTYPE_BOOL	1		
Revenue Growth	RevenueGrowth	DTYPE_NUMBER	22		
Service Provider Flag	ServiceProviderFlag	DTYPE_TEXT	1		
Status	Status	DTYPE_TEXT	30		Bound by FINS_COMPETITOR_STATUS_MLOV
Total # Meters	TotalMeters	DTYPE_NUMBER	22		
Total Potential Volume	TotalPotentialVolume	DTYPE_CURRENCY	22		
Type	Type	DTYPE_TEXT	30	✓	Bound by ACCOUNT_TYPE
Unionized	Unionized	DTYPE_BOOL	1		
YTD Net Contribution	YTDNetContribution	DTYPE_CURRENCY	22		
operation	operation	DTYPE_TEXT	30		

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Id	Id	DTYPE_ID	150		For cross-referencing
Account_Business Address					
Address Id	AddressId	DTYPE_ID	30	✓	
Address Integration Id	AddressIntegrationId	DTYPE_TEXT	30		
Bill Address Flag	BillAddressFlag	DTYPE_TEXT	1		
City	City	DTYPE_TEXT	50		
Country	Country	DTYPE_TEXT	50		Use for pick a related entity
Email Address	EmailAddress	DTYPE_TEXT	50		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Postal Code	PostalCode	DTYPE_TEXT	30		
Ship Address Flag	ShipAddressFlag	DTYPE_TEXT	1		
State	State	DTYPE_TEXT	10		Bound by STATE_ABBREV
Street Address	StreetAddress	DTYPE_TEXT	200		
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
Action					
Alarm	Alarm	DTYPE_BOOL	1		
Assignment Excluded	AssignmentExcluded	DTYPE_BOOL	1		
Associated Cost	AssociatedCost	DTYPE_CURRENCY	22		
Billable Flag	BillableFlag	DTYPE_BOOL	1		
Call Id	CallId	DTYPE_TEXT	30		
Category	Category	DTYPE_TEXT	30	✓	Use for picking a related entity
Class	Class	DTYPE_TEXT	30		Bound by FS_ACTIVITY_CLASSES

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Comment	Comment	DTYPE_TEXT	"1,500"		
Description	Description	DTYPE_TEXT	100		
Display	Display	DTYPE_TEXT	30		
Done	Done	DTYPE_UTCDATETIME	7		
Done Flag	DoneFlag	DTYPE_BOOL	1		
Due	Due	DTYPE_UTCDATETIME	7		
Duration Minutes	DurationMinutes	DTYPE_INTEGER	22		Bound by APPOINTMENT_DURATION
Expense Related Flag	ExpenseRelatedFlag	DTYPE_BOOL	1		
Id	Id	DTYPE_TEXT	100	✓	
Integration Id	IntegrationId	DTYPE_TEXT	30		
No Sooner Than Date	NoSoonerThanDate	DTYPE_UTCDATETIME	7		
Objective	Objective	DTYPE_TEXT	30		
Owned By	OwnedBy	DTYPE_TEXT	30	✓	Use for pick a related entity
Owned By Id	OwnedById	DTYPE_TEXT	30	✓	
Primary Owned By	PrimaryOwnedBy	DTYPE_TEXT	50		Use for pick a related entity
Primary Owner Id	PrimaryOwnerId	DTYPE_ID	30		
Priority	Priority	DTYPE_TEXT	30		Bound by ACTIVITY_PRIORITY
Private	Private	DTYPE_BOOL	1		
Repeating	Repeating	DTYPE_BOOL	1		
Repeating Expires	RepeatingExpires	DTYPE_DATETIME	7		
Status	Status	DTYPE_TEXT	30		Bound by EVENT_STATUS

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Type	Type2	DTYPE_TEXT	30	✓	Bound by TODO_TYPE
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
Contact					
Account Id	AccountId	DTYPE_ID	30		
Account Integration Id	AccountIntegrationId	DTYPE_TEXT	30		
Assistant	Assistant	DTYPE_TEXT	50		
Assistant Phone #	AssistantPhone	DTYPE_PHONE	40		
Cellular Phone #	CellularPhone	DTYPE_PHONE	40		
Fax Phone #	FaxPhone	DTYPE_PHONE	40		
First Name	FirstName	DTYPE_TEXT	50	✓	
Home Phone #	HomePhone	DTYPE_TEXT	40		
Job Title	JobTitle	DTYPE_TEXT	75		
Last Name	LastName	DTYPE_TEXT	50	✓	
M/M	MM	DTYPE_TEXT	15		
Middle Name	MiddleName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100	✓	
Person UId	PersonUId	DTYPE_ID	30	✓	
Preferred Communication Method	PreferredCommunicationMethod	DTYPE_TEXT	20		Bound by COMM_METHOD
Privacy Code	PrivacyCode	DTYPE_TEXT	30	✓	Bound by PRIVACY_CODE
Row Id	RowId	DTYPE_ID	100		
Work Phone #	WorkPhone	DTYPE_PHONE	45		
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		

FINCORP Deposit Account

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Account Branch Number	AccountBranchNumber	DTYPE_TEXT	100		
Account Category	AccountCategory	DTYPE_TEXT	30		Bound by FINS_ASSET_PROD_CLASS_MLOV
Account Number	AccountNumber	DTYPE_TEXT	100	✓	
Account Status	AccountStatus	DTYPE_TEXT	30		Bound by FINS_CRPST_LFLN_IPFS_PLC_MLOV
Branch	Branch	DTYPE_TEXT	50		
Current Balance	CurrentBalance	DTYPE_CURRENCY	22		
Id	Id	DTYPE_TEXT	100		
Integration Id	Integration Id	DTYPE_TEXT	30		
Name	Name	DTYPE_TEXT	100	✓	Use for pick a related entity
Product Id	ProductId	DTYPE_ID	100	✓	
Type	Type	DTYPE_TEXT	30		Bound by FINS_OWNERSHIP_TYPE_MLOV
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
FINCORP Loan Account					
Account Branch Number	AccountBranchNumber	DTYPE_TEXT	30		
Account Category	AccountCategory	DTYPE_TEXT	30		Bound by FINS_ASSET_PROD_CLASS_MLOV
Account Number	AccountNumber	DTYPE_TEXT	100	✓	
Account Status	AccountStatus	DTYPE_TEXT	30		Bound by FINS_CRPST_LFLN_IPFS_PLC_MLOV

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Current Balance	CurrentBalance	DTYPE_CURRENCY	22		
Id	Id	DTYPE_TEXT	30		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Name	Name	DTYPE_TEXT	100	✓	Use for pick a related entity
Product Id	ProductId	DTYPE_ID	100	✓	Use for pick a related entity
Type	Type	DTYPE_TEXT	30		FINS_OWNERSHIP_TYPE_MLOV
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
Related Industry					
Industry	Industry	DTYPE_TEXT	50	✓	
Industry Id	IndustryId	DTYPE_ID	30	✓	
Industry Integration Id	IndustryIntegrationId	DTYPE_TEXT	30		
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
SIC Code	SICCode	DTYPE_TEXT	30	✓	
Related Organization					
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Organization	Organization	DTYPE_TEXT	100	✓	
Organization Id	OrganizationId	DTYPE_ID	100	✓	
Organization Integration Id	OrganizationIntegrationId	DTYPE_TEXT	30		
Related Sales Reps					
Division	Division	DTYPE_TEXT	255	✓	Use to pick a related entity
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Login	Login	DTYPE_ID	15		
Position	Position	DTYPE_TEXT	50		
Position Id	PositionId	DTYPE_ID	30		

Table 113. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
Position Integration Id	PositionIntegrationId	DTYPE_TEXT	30		
Sub Account					
Account Status	AccountStatus2	DTYPE_TEXT	30		Bound by ACCOUNT_STATUS
Id	Id	DTYPE_ID	100	✓	
Location	Location	DTYPE_TEXT	50		
Main Phone Number	MainPhoneNumber	DTYPE_PHONE	40		
Name	Name	DTYPE_TEXT	100	✓	
Party UId	PartyUId	DTYPE_TEXT	30	✓	
Type	Type	DTYPE_TEXT	255	✓	Bound by ACCOUNT_TYPE
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		

Cross Reference

UCM Cross Reference messages are used in query and response operations for account, contact and household cross references. There is a pair of request and response messages for each object. Figure 29 describes the structure of these messages.

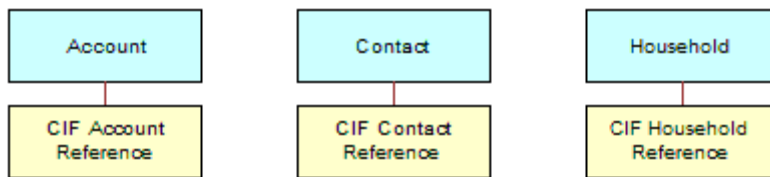


Figure 29. UCM Cross Reference Structure

Table 114. Account Cross Reference Integration Objects

Integration Component	Business Component	XML Tague	User Keys
UCMAccountExternalID	Contact	LookUpAccountExternalID	UK1:PartyUIDUK2:Id
UCMAccountExternalIDRs	Contact	LookUpAccountExternalIDResponse	UK1:PartyUIDUK2:Id

Table 115. Contact Cross Reference Integration Objects

Integration Component	Business Component	XML Tague	User Keys
UCMContactExternalID	Contact	LookUpContactExternalID	UK1:PartyUIDUK2:Id
UCMContactExternalIDRs	Contact	LookUpContactExternalIDResponse	UK1:PartyUIDUK2:Id

Table 116. Household Cross Reference Integration Objects

Integration Component	Business Component	XML Tague	User Keys
UCMHouseholdExternalID	Household	LookUpHouseholdExternalID	UK1:IdUK2:Integrati onIdUK3:PartyUID
UCMHouseholdExternalIDRs	Household	LookUpHouseholdExternalID Response	UK1:IdUK2:Integrati onIdUK3:PartyUID

Table 117. Account Cross Reference Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Note
Account					
Account Status	AccountStatus	DTYPE_TEXT	24		
Account Type Code	AccountTypeCode	DTYPE_TEXT	30		
Annual Revenue	AnnualRevenue	DTYPE_TEXT			
Client Flag	ClientFlag	DTYPE_TEXT			
Competitor	Competitor	DTYPE_TEXT			
Conflict Id	ConflictId	DTYPE_ID	30		
Created	Created	DTYPE_DATETIME	30		
Credit Agency Number	CreditAgencyNumber	DTYPE_TEXT	30		
Currency Code	CurrencyCode	DTYPE_TEXT	15		
Facility Flag	FacilityFlag	DTYPE_TEXT			
Federal Tax Number	FederalTaxNumber	DTYPE_TEXT	15		
Home Page	HomePage	DTYPE_TEXT	100		
Id	Id	DTYPE_ID	30		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Internal Org Flag	InternalOrgFlag	DTYPE_TEXT			
Investor Flag	InvestorFlag	DTYPE_TEXT			
Legal Status	LegalStatus	DTYPE_TEXT	30		
Location	Location	DTYPE_TEXT	50		
Main Fax Number	MainFaxNumber	DTYPE_TEXT			
Main Phone Number	MainPhoneNumber	DTYPE_TEXT			
Master Account Id	MasterAccountId	DTYPE_ID			
Mod Id	ModId	DTYPE_ID	30		
Name	Name	DTYPE_TEXT	100		
Partner Flag	PartnerFlag	DTYPE_TEXT			
Party UId	PartyUId	DTYPE_TEXT	100		
Service Provider Flag	ServiceProviderFlag	DTYPE_TEXT			
Status	Status	DTYPE_TEXT	30		
Type	Type	DTYPE_TEXT	30		
Updated	Updated	DTYPE_DATETIME	30		

Table 117. Account Cross Reference Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Note
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
CIF Account Reference					
Account Id	AccountId	DTYPE_ID		✓	
CIF System Id	CIFSystemId	DTYPE_ID		✓	
Comment	Comment	DTYPE_TEXT	250		
Conflict Id	ConflictId	DTYPE_ID	30		
Created	Created	DTYPE_DATETIME	30		
External Id1	ExternalId1	DTYPE_TEXT	100		
External Id2	ExternalId2	DTYPE_TEXT	100		
External Id3	ExternalId3	DTYPE_TEXT	100		
Id	Id	DTYPE_ID	30		
Mod Id	ModId	DTYPE_ID	30		
System Name	SystemName	DTYPE_TEXT	100		
System Number	SystemNumber	DTYPE_TEXT	30		
Updated	Updated	DTYPE_DATETIME	30		
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		

Table 118. Contact Cross Reference Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Note
Contact					
Cellular Phone #	CellularPhone	DTYPE_TEXT			
Conflict Id	ConflictId	DTYPE_ID	30		
Created	Created	DTYPE_DATETIME	30		
Date of Birth	DateofBirth	DTYPE_DATETIME			
Email Address	EmailAddress	DTYPE_TEXT	50		
Fax Phone #	FaxPhone	DTYPE_TEXT			
First Name	FirstName	DTYPE_TEXT	50		
Home Phone #	HomePhone	DTYPE_TEXT			
Id	Id	DTYPE_ID	30		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Job Title	JobTitle	DTYPE_TEXT	75		
Last Name	LastName	DTYPE_TEXT	50		
M/F	MF	DTYPE_TEXT	30		
M/M	MM	DTYPE_TEXT	15		
Marital Status	MaritalStatus	DTYPE_TEXT	30		
Middle Name	MiddleName	DTYPE_TEXT	50		
Mod Id	ModId	DTYPE_ID	30		
Mother Maiden Name	MotherMaidenName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100		
Person UId	PersonUId	DTYPE_TEXT	100		
Social Security Number	SocialSecurityNumber	DTYPE_TEXT			
Status	Status	DTYPE_TEXT	30		
Updated	Updated	DTYPE_DATETIME	30		
Work Phone #	WorkPhone	DTYPE_TEXT			
Work Phone Extension	WorkPhoneExtension	DTYPE_TEXT	10		
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
CIF Contact Reference					
CIF System Id	CIFSystemId	DTYPE_ID		✓	

Table 118. Contact Cross Reference Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Note
Comment	Comment	DTYPE_TEXT	250		
Conflict Id	ConflictId	DTYPE_ID	30		
Contact Id	ContactId	DTYPE_ID		✓	
Created	Created	DTYPE_DATETIME	30		
External Id1	ExternalId1	DTYPE_TEXT	100		
External Id2	ExternalId2	DTYPE_TEXT	100		
External Id3	ExternalId3	DTYPE_TEXT	100		
Id	Id	DTYPE_ID	30		
Mod Id	ModId	DTYPE_ID	30		
System Name	SystemName	DTYPE_TEXT	100		
System Number	SystemNumber	DTYPE_TEXT	30		
Updated	Updated	DTYPE_DATETIME	30		
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		

Table 119. Household Cross Reference Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Note
Household					
Category	Category	DTYPE_TEXT	30		
Conflict Id	ConflictId	DTYPE_ID	30		
Created	Created	DTYPE_DATETIME	30		
Currency Code	CurrencyCode	DTYPE_TEXT	20		
Fax Number	FaxNumber	DTYPE_TEXT			
Household Name	HouseholdName	DTYPE_TEXT	50		
Household Number	HouseholdNumber	DTYPE_TEXT	100		
Id	Id	DTYPE_ID	30		
Integration Id	IntegrationId	DTYPE_TEXT	30		
Mod Id	ModId	DTYPE_ID	30		
Party UId	PartyUId	DTYPE_TEXT	100		
Phone Number	PhoneNumber	DTYPE_TEXT			

Table 119. Household Cross Reference Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Note
Status	Status	DTYPE_TEXT	30		
Type	Type	DTYPE_TEXT	30		
Updated	Updated	DTYPE_DATETIME	30		
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		
CIF Household Reference					
CIF System Id	CIFSystemId	DTYPE_ID		✓	
Comment	Comment	DTYPE_TEXT	250		
Conflict Id	ConflictId	DTYPE_ID	30		
Household Id	ContactId	DTYPE_ID		✓	
Created	Created	DTYPE_DATETIME	30		
External Id1	ExternalId1	DTYPE_TEXT	100		
External Id2	ExternalId2	DTYPE_TEXT	100		
External Id3	ExternalId3	DTYPE_TEXT	100		
Id	Id	DTYPE_ID	30		
Mod Id	ModId	DTYPE_ID	30		
System Name	SystemName	DTYPE_TEXT	100		
System Number	SystemNumber	DTYPE_TEXT	30		
Updated	Updated	DTYPE_DATETIME	30		
operation	operation	DTYPE_TEXT	30		
searchspec	searchspec	DTYPE_TEXT	250		

A

Additional License Restrictions

Siebel Master Data Applications are licensed subject to additional license grants and restrictions, see [“Additional License Grants and Restrictions Regarding Siebel Master Data Applications.”](#) These restrictions supersede all other grants and restrictions.

Additional License Grants and Restrictions Regarding Siebel Master Data Applications

Siebel Master Data Applications are licensed to 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 Master Data Applications without purchase of the Customer Data Steward or Product Data Steward modules;
- 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 a source of customer, product, activity, sales, service, marketing, or field service information and any extensions made through the use of Siebel Tools;
- Customer shall have the right to use the Siebel Tools to configure the Siebel Master Data Applications solely in accordance with the Documentation to configure only those Business Objects and Business Components (BusComps) included with the licensed Siebel Master Data Applications (as set forth in Appendix 1) through the Object Manager and Enterprise Application Integration (EAI), Application Services Interfaces (ASIs), Enterprise Integration Manager (EIM), and Business Process Designer.
- Customer shall have the right to use System Administration solely in accordance with the Documentation including Business Process Designer and EIM administration and EAI configuration;
- Customer shall have the right to have programmatic access only to those Business Objects and BusComps included with the licensed Siebel Master Data Applications, solely through Object Manager and EAI, EIM, and 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 Siebel Master Data Applications (including the Siebel Data Model) for Siebel non-CRM (Customer Relationship Management) applications.

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

Table 120 sets forth the Business Components included in Siebel Master Data Applications.

Table 120. Business Components Included in Siebel Master Data Applications

Master Data Module	Business Components
Siebel Universal Customer Master	Party
	Party Contact
	Party Relationship To
	Account
	Contact
	Contact Relationship
	Household
	Business Address
	Personal Address
	Channel Partner
	Partnership Contact
	Opportunity
	Position
	Employee
	Expense
	FINCORP Account (Asset)
	FINCORP Deposit Account
	FINCORP Loan Account
	FINS AG Agent
	FINS All Policy Details
	FINS Group Policy
	FINS Household Policies (Read Only)
FINS Life Policy	
Household FIN Accounts (Read Only)	
INS All Policies	
INS Policy (Asset)	

Table 120. Business Components Included in Siebel Master Data Applications

Master Data Module	Business Components
Siebel Universal Activity Master	Action
	Contact Note
	Contact Private Note
	Personal Payment Profile
	Account Profile Attributes
	Account Credit Profile
	Account Note
	Account Private Note
	Account External Product
	Asset Mgmt (Asset)
	Sub Account
	Entitlement Account
	Entitlement Contact
	Service Agreement
	Coverage Attributes
	Partner Certification
	Action Copy
	Coverage Attributes
	Household Action (Read Only)
	Household Contact Note (Read Only)
	Household Service Agreement (Read Only)
	Partner Certification Criteria
	Partner Product
	Product Line
	FINCORP Client Contact Management
	FIN Contact Income
	FIN Contact Employment
	FINS Financial Statement Contact
	FINS Financial Accounts for Contacts
	FINCORP Contact Education
FINCORP Contact Experience	

Table 120. Business Components Included in Siebel Master Data Applications

Master Data Module	Business Components
Siebel Universal Activity Master (continued)	FINCORP Investor
	FINCORP Transaction
	FINS AG Agency's Contracts
	FINS AG Agency's Errors And Omissions
	FINS AG Agency's Licenses And Appointments
	FINS AG Agent's Contracts
	FINS AG Agent's Errors And Omissions
	FINS AG Agent's Licenses And Appointments
	FINS AG Agent's NASD Registration
	FINS Billing Accounts
	FINS Billing Accounts (No Contact Copy)
	FINS Contact Household Contact
	FINS Household Billing Accounts (Read Only)

Table 120. Business Components Included in Siebel Master Data Applications

Master Data Module	Business Components
Siebel Universal Product Master	Complex Product
	Internal Product
	Internal Product Attributes
	Internal Product Attachment
	Internal Product Image
	Internal Product Translation
	Component Product
	Price List
	Price List Item
	Price Book
	Price Book Attributes
	Price Book Attribute Values
	Price Book Groups
	Admin Product Line
	Admin Product Line Fee
	Consumer Product
	Catalog Category
	Product Entitlement Details
	Key Feature
	Admin Product Comparison
	Sales Tool by Product
	XA Attribute Value
	Cost List Item
	Product Defect
	Partner Product
	Revenue

Table 120. Business Components Included in Siebel Master Data Applications

Master Data Module	Business Components
Siebel Universal Sales Master	Opportunity
	Opportunity (Orders)
	Competitor
	Competitor 2
	Quote
	Quote Attachment
	Quote Item
	Orders
	Order Entry (Orders)
	Order Entry (Line Items)
	Order Entry (Order Terms)
	Decision Issue
	Decision Issue Attachment
	Opportunity Forecast
	Opportunity Forecast Detail (Wide)
	Opportunity Forecast Line
	Opportunity Person Forecast
	Opportunity Product Forecast

Table 120. Business Components Included in Siebel Master Data Applications

Master Data Module	Business Components
Siebel Universal Service Master	Service Request
	Related Service Request
	Solution
	Related Solution
	Resolution Activity
	SR Resolution Item
	RMA
	Service Agreement Account
	Service Agreement
	Service Agreement Contact
	Service Agreement Product
	Service Agreement Attachment
	Product Defect
	Decision Issue
	Customer Survey
	Customer Product
	Activity Plan
	Related Service Request
FIN Service Request	

Table 120. Business Components Included in Siebel Master Data Applications

Master Data Module	Business Components
Siebel Universal Marketing Master	Campaign
	Campaign Achievement
	Campaign Contact
	Campaign Groups
	Campaign List Contact
	Campaign Lists
	Campaign Occurrence Offer
	Campaign Recipient
	Prospect
	Contact-Prospect Campaigns
	Prospect Partner
	Prospect Recipients
	Prospect Substitution
	List Management
	List Mgmt List Member Prospect
	List Mgmt Prospective Contact
	List Mgmt Prospective Contact Mkt Segment
	List Mgmt Prospective Contact (Attributes)
	Offer
	Response
	Segments
	Programs
	Program Container
Program Container Occurrences	
Program Occurrences	

Table 120. Business Components Included in Siebel Master Data Applications

Master Data Module	Business Components
Siebel Universal Field Service Master	FS Activities (Time, Expense, Material, Skills, Steps, Instructions)
	Service Agreement
	Entitlement
	Warranty
	Asset Management
	Asset Measurement
	Preventive Maintenance
	Invoice
	Inventory
	Order entry
	Cycle Counting
	Shipping
	Receiving
	Replenishment
	Repair
	FS Price List
FS Product	
Scheduling	
Siebel Universal Identification Master	Contact
	CIF Contact Reference
	Account
	CIF Account Reference

The Siebel Master Data Applications are licensed based on:

- The number of unique Contact Records which the customer may store in the Universal Customer Master.
- The number of Application Instances integrated to the Siebel Universal Customer Master.

A Contact Record is a database record of an individual contact that is stored in the Siebel Data Model. For those customers who cannot identify or otherwise count the number of individual contacts for which the application will provide master functionality, on an exception basis the products may be priced and licensed on the basis of the number of account records (records in the Siebel Data Model) in the Siebel Universal Customer Master deployment, using a ratio that assumes 20 contact records for each account record. Approval is required to price and license on this basis.

An Application Instance is each individual installation of an application. All Application Instances that meet the following criteria are counted as Application Instances that are integrated to the Siebel Master Data Applications:

- Access information from Siebel Master Data Applications as a source of information.
- Send information to Siebel Master Data Applications.
- Receive information from Siebel Master Data Applications.

Such Application Instances can be integrated to Siebel Master Data Applications by means of direct connection, integration server, a messaging bus, flat file, or any other means of real-time, near real-time or batch integration. Application Instances used solely for inter-application communication, such as IBM MQ Series, are not counted as Application Instances integrated to Siebel Master Data Applications.

The Siebel Universal Product Master Applications are licensed based on:

- The number of unique Product Records which the customer may store in the Universal Product Master.
- The number of Application Instances integrated to the Siebel Universal Product Master.

A Product Record is a database record of an individual product that is stored in the Siebel Data Model.

Index

A

Access Control, enforcing 19

Account Activity messages

about 125

Account Activity Delete Integration
Components 126

Account Activity Insert Integration
Components 126

Account Activity Integration Component
Fields 126

Account Activity Query by Name Integration
Components 125

Account Activity Query Integration
Components 125

Account Activity Update Integration
Components 125

Account Address messages

about 120

Account Address Delete Integration
Components 121

Account Address Insert Integration
Components 121

Account Address Integration Component
Fields 121

Account Address Query by Name Integration
Components 120

Account Address Query Integration
Components 120

Account Address Update Integration
Components 120

Account messages

about 146

Account Delete Integration
Components 147

Account Insert Integration
Components 147

Account Integration Component Fields 147

Account Query by Name Integration
Components 146

Account Query Integration
Components 146, 156

Account Update Integration
Components 146

Account Product messages

about 132

Account Product Delete Integration
Components 133

Account Product Insert Integration
Components 133

Account Product Integration Component
Fields 133

Account Product Query by Name Integration
Components 132

Account Product Query Integration
Components 132

Account Product Update Integration
Components 132

Account Profile messages

about 116

Account Profile Integration Component
Fields 117

Account Profile Query by Name Integration
Components 116

Account Profile Query Integration
Components 116

Account Profile Update Integration
Components 116

Account Reference record, creating 55

Account Relationship messages

about 138

Account Relationship Delete Integration
Components 139

Account Relationship Insert Integration
Components 139

Account Relationship Integration Component
Fields 139

Account Relationship Query by Name
Integration Components 138

Account Relationship Query Integration
Components 138

Account Relationship Update Integration
Components 138

administering Siebel UCM Applications 19

administration

Account Reference record, creating 55

Contact Reference record, creating 56

cross-referencing customer
identification 55

deleting a registered system 48

Household Reference record, creating 57

registering systems connected with UCM
applications 47

Siebel UCM Application integration objects,
about 57

- Siebel UCM Application integration objects,
 - modifying 58
- Siebel UCM Applications integration
 - services 60
- system privileges, setting 49
- System Publish and Subscribe,
 - configuring 50
- UCM dispatcher map integration objects,
 - about 59
- UCM dispatcher map integration objects,
 - viewing user properties 59
- UCM envelope integration objects,
 - modifying 58
- architecture**
 - about 21
 - about and diagram 17
 - administering Siebel UCM Applications 19
 - database tables, functions and services 18
 - integration services, about 20
 - Siebel Connector and integration 31
 - UCM business objects layer 22
 - UCM Data Objects layer 30
 - UCM user interface layer 22
- B**
- business objects layer**
 - about 22
 - object manager and data manager 22
 - Siebel Industry eAI Architecture 22
- business process flow, support of Siebel Industry eAI architecture** 24
- business services, support of Siebel Industry eAI architecture** 26
- C**
- check mark in Subscribe column** 50
- CIFOperation_GetValue, about and user properties** 61
- CIFOperation_Query** 61
- CIFOperation_XMLQuery** 61
- communication**
 - inbound communication and UCM Converter 35
- components** 10
- Contact Activity messages**
 - about 84
 - Contact Activity Delete Integration Components 85
 - Contact Activity Insert Integration Components 84
 - Contact Activity Integration Component Fields 85
 - Contact Activity Query by Name Integration Components 84
 - Contact Activity Query Integration Components 84
 - Contact Activity Update Integration Components 84
- Contact Address messages**
 - about 81
 - Contact Address Delete Integration Components 82
 - Contact Address Insert Integration Components 81
 - Contact Address Integration Component Fields 82
 - Contact Address Query by Name Integration Components 81
 - Contact Address Query Integration Components 81
 - Contact Address Update Integration Components 81
- Contact messages**
 - about 97
 - Contact Delete Integration Components 98
 - Contact Integration Component Fields 98
 - Contact Query by Name Integration Components 97
 - Contact Query Integration Components 97
 - Contact Update Integration Components 97
- Contact Product messages**
 - about 88
 - Contact Product Delete Integration Components 89
 - Contact Product Insert Integration Components 88
 - Contact Product Integration Component Fields 89
 - Contact Product Query by Name Integration Components 88
 - Contact Product Query Integration Components 88
 - Contact Product Update Integration Components 88
- Contact Profile messages**
 - about 77
 - Contact Profile Delete Integration Components 78
 - Contact Profile Insert Integration Components 77
 - Contact Profile Integration Component Fields 78
 - Contact Profile Query by Name Integration Components 77
 - Contact Profile Query Integration Components 77

- Contact Profile Update Integration Components 77
- Contact Reference record, creating** 56
- Contact Relationship messages**
 - about 92
 - Contact Relationship Delete Integration Components 93
 - Contact Relationship Insert Integration Components 93
 - Contact Relationship Integration Component Fields 93
 - Contact Relationship Query by Name Integration Components 92
 - Contact Relationship Query Integration Components 92
 - Contact Relationship Update Integration Components 92
- CRMML standards**
 - about and XML messages 20
 - body section 30
 - CRMML syntax and rules 26
 - envelope and header information, about 72
 - envelope and header section 27
 - Siebel Industry eAI architecture, support of 26
- Cross Reference messages**
 - about 156
- customer identification**
 - Account Reference record, creating 55
 - Contact Reference record, creating 56
 - cross-referencing, about 55
 - Household Reference record, creating 57
- customer information**
 - Siebel Universal Activity Master, about using 8
- Customer Relationship Management Markup Language**
 - See CRMML standard
- D**
- D&B information, about** 11
- Daily Batch publish types**
 - about 44
 - argument types 45
- data manager layer, about** 22
- database auto-reconnect, about** 21
- database tables**
 - functions and services 18
- Delete functionality** 18
- deleting, a registered system deployment**
 - sharing database with eBusiness application scenario 15
 - stand-alone application scenario 14
 - UCM works with eBusiness and back-office applications 15
- E**
- Entity Relationship Designer** 10
- entity relationship diagrams (ERDs), about creating** 10
- envelope integration objects, modifying** 58
- exporting**
 - See Siebel Enterprise Integration Manager
- F**
- fault tolerance, designed for** 21
- features, new** 5
- field service**
 - storing organization's field service information 9
- framework**
 - administering Siebel UCM Applications architecture and diagram 19
 - database tables, functions and services 18
 - integration services 20
 - Siebel Connector and integration 31
 - UCM business objects layer 22
 - UCM Data Objects layer 30
 - UCM user interface layer 22
- G**
- grants and restrictions** 163
- H**
- Household messages**
 - about 107
 - Household Delete Integration Components 108
 - Household Insert Integration Components 107
 - Household Integration Component Fields 108
 - Household Query by Name Integration Components 107
 - Household Query Integration Components 107
 - Household Update Integration Components 107
- Household Reference record, creating** 57
- I**
- importing**

See Siebel Enterprise Integration Manager

inbound communication, and UCM Converter 35

inbound data flow, support of Siebel Industry eAI architecture 24

Inbound HTTP Transport, configuring the Siebel Web Engine 64

inbound messages, and UCM Dispatcher 41

inbound server administration

- about 62
- MQSeries Service Receiver parameters, configuring 64
- named subsystem, configuring 63

Insert functionality 18

integration

- Siebel Connector, and 31
- Siebel EIM and integration 46
- Transport Adapter 46
- UCM Converter user properties 35
- UCM Dispatcher user properties 41
- UCM Publish/Subscribe Service methods 44
- UCM Security Manager user properties 42
- UCM Transaction Manager 31

Integration Object Builder, about 25

integration objects

- about 57
- modifying 58
- Siebel Industry eAI architecture, support of 25
- Siebel UCM Applications integration services 60
- UCM dispatcher map integration objects 59
- UCM dispatcher map integration objects, viewing user properties 59
- UCM envelope integration objects, modifying 58

integration services

- about and Siebel Connector 20
- architecture, about 21

K

key maps

- Account Reference record, creating 55
- Contact Reference record, creating 56
- creating, about 55
- Household Reference record, creating 57

L

license 163

M

marketing

- storing organization's marketing information 9

master file modules

See modules

messages

- about 76
- Account Address 120
- application service interfaces supported 67
- CRMML, about envelope and header information 72
- inbound messages and UCM Dispatcher 41
- UCM Account Activity messages 125
- UCM Account messages 146
- UCM Account Product messages 132
- UCM Account Profile messages 116
- UCM Account Relationship messages 138
- UCM Contact Activity messages 84
- UCM Contact Address messages 81
- UCM Contact messages 97
- UCM Contact Product messages 88
- UCM Contact Profile messages 77
- UCM Contact Relationship messages 92
- UCM Cross Reference messages 156
- UCM Household messages 107
- UCM Personal Address messages 112
- universally unique identifier 76

Method Security Check arguments 44

modules

- about 8
- Siebel Universal Activity Master 8
- Siebel Universal Customer Master, about 8
- Siebel Universal Customer Master, as base module 7
- Siebel Universal Field Service Master 9
- Siebel Universal Identification Master 10
- Siebel Universal Marketing Master 9
- Siebel Universal Product Master 9
- Siebel Universal Sales Master 9
- Siebel Universal Service Master 9

MQSeries Server Receiver parameter, configuring 64

N

named subsystem, configuring 63

native SQL, and UCM Data Objects layer 30

new features 5

O

object manager layer, about 22

P**Personal Address messages**

- about 112
- FINCORP Account Integration Component Fields 113
- FINCORP Account Query Integration Components 113
- Personal Address Integration Component Fields 112
- Personal Address Query Integration Components 112
- UCM FINCORP 113

product components 10**products**

- storing organization product's information 9

Publish services, about 18**Q****Query functionality** 18**R****Real Time publish types**

- about 44
- argument types 45

registering systems

- connected with UCM applications 47
- deleting a registered system 48

Request and Reply services, about restrictions 163**S****sales**

- storing organization's sales information 9

scenarios 7

- Siebel Universal Customer Master scenario 1 13
- Siebel Universal Customer Master scenario 2 14

service

- storing organization's service information 9

Siebel Analytics 11**Siebel Application Object Manager** 10**Siebel Assignment Manager** 11**Siebel Audit Trail Functionality** 11**Siebel Business Process Designer** 10**Siebel Business Process Designer, about and business data flow** 25**Siebel Connector**

- about and integration 31
- Siebel EIM and integration 46

- Transport Adapter 46
- UCM Converter user properties 35
- UCM Dispatcher user properties 41
- UCM Publish/Subscribe Service methods 44
- UCM Security Manager user properties 42
- UCM Transaction Manager 31

Siebel Connector, about using 20**Siebel Connectors**

- Siebel Industry eAI architecture, support of 22

Siebel D&B Integration 11**Siebel Data Quality** 11**Siebel eBusiness Application Integration** 10**Siebel EIM, and integration** 46**Siebel Enterprise Integration Manager** 10**Siebel Entity Relationship Designer** 10**Siebel Industry eAI architecture**

- about 22
- business process flow 24
- business services 26
- CRMML standard 26
- inbound data flow 24
- integration objects 25
- workflow integration 25

Siebel Replication Manager 11**Siebel Tools** 11**Siebel UCM Applications integration objects**

- about 57
- modifying 58
- modifying UCM envelope integration objects 58

Siebel UCM Applications integration services

- about 60
- inbound server administration 62
- Siebel Universal Master Application, configuring 65
- Siebel Web Engine, configuring to run Inbound HTTP Transport 64
- UCM Transaction Manager 60

Siebel Universal Activity Masters 8**Siebel Universal Customer Master**

- contents of 7
- module, about 8
- scenario 1 13
- scenario 2 14

Siebel Universal Field Service Master 9**Siebel Universal Identification Master** 10**Siebel Universal Marketing Master** 9**Siebel Universal Master application**

- Extension Service, configuring to run 65

Siebel Universal Sales Master 9**Siebel Universal Service Master** 9

Siebel Web Engine

Inbound HTTP Transport, configuring 64

SQL, and UCM Data Objects layer 30**stand-alone application scenario** 14**Subscribe column, check mark in system privileges, setting** 49**System Publish and Subscribe**

configuring 50

T**third-party solutions**

See Siebel eBusiness Application Integration

Transport Adapter, and Siebel**Connector** 46**U****UCM Account Activity messages**

about 125

Account Activity Delete Integration
Components 126Account Activity Insert Integration
Components 126Account Activity Integration Component
Fields 126Account Activity Query by Name Integration
Components 125Account Activity Query Integration
Components 125Account Activity Update Integration
Components 125**UCM Account Address messages**

about 120

Account Address Delete Integration
Components 121Account Address Insert Integration
Components 121Account Address Integration Component
Fields 121Account Address Query by Name Integration
Components 120Account Address Query Integration
Components 120Account Address Update Integration
Components 120**UCM Account messages**

about 146

Account Delete Integration
Components 147Account Insert Integration
Components 147

Account Integration Component Fields 147

Account Query by Name Integration
Components 146

Account Query Integration

Components 146, 156

Account Update Integration

Components 146

UCM Account Product messages

about 132

Account Product Delete Integration
Components 133Account Product Insert Integration
Components 133Account Product Integration Component
Fields 133Account Product Query by Name Integration
Components 132Account Product Query Integration
Components 132Account Product Update Integration
Components 132**UCM Account Profile messages**

about 116

Account Profile Integration Component
Fields 117Account Profile Query by Name Integration
Components 116Account Profile Query Integration
Components 116Account Profile Update Integration
Components 116**UCM Account Relationship messages**

about 138

Account Relationship Delete Integration
Components 139Account Relationship Insert Integration
Components 139Account Relationship Integration Component
Fields 139Account Relationship Query by Name
Integration Components 138Account Relationship Query Integration
Components 138Account Relationship Update Integration
Components 138**UCM business objects layer**

about 22

object manager and data manager 22

Siebel Industry eAI architecture 22

UCM Contact Activity messages

about 84

Contact Activity Delete Integration
Components 85Contact Activity Insert Integration
Components 84Contact Activity Integration Component
Fields 85

- Contact Activity Query by Name Integration Components 84
- Contact Activity Query Integration Components 84
- Contact Activity Update Integration Components 84
- UCM Contact Address messages**
 - about 81
 - Contact Address Delete Integration Components 82
 - Contact Address Insert Integration Components 81
 - Contact Address Integration Component Fields 82
 - Contact Address Query by Name Integration Components 81
 - Contact Address Query Integration Components 81
 - Contact Address Update Integration Components 81
- UCM Contact messages**
 - about 97
 - Contact Delete Integration Components 98
 - Contact Integration Component Fields 98
 - Contact Query by Name Integration Components 97
 - Contact Query Integration Components 97
 - Contact Update Integration Components 97
- UCM Contact Product messages**
 - about 88
 - Contact Product Delete Integration Components 89
 - Contact Product Insert Integration Components 88
 - Contact Product Integration Component Fields 89
 - Contact Product Query by Name Integration Components 88
 - Contact Product Query Integration Components 88
 - Contact Product Update Integration Components 88
- UCM Contact Profile messages**
 - about 77
 - Contact Profile Delete Integration Components 78
 - Contact Profile Insert Integration Components 77
 - Contact Profile Integration Component Fields 78
 - Contact Profile Query by Name Integration Components 77
 - Contact Profile Query Integration Components 77
- Contact Profile Update Integration Components 77
- UCM Contact Relationship messages**
 - about 92
 - Contact Relationship Delete Integration Components 93
 - Contact Relationship Insert Integration Components 93
 - Contact Relationship Integration Component Fields 93
 - Contact Relationship Query by Name Integration Components 92
 - Contact Relationship Query Integration Components 92
 - Contact Relationship Update Integration Components 92
- UCM Converter**
 - argument specifications 37
 - methods 36
 - user properties 35
- UCM Converter, about and user properties** 61
- UCM Cross Reference messages**
 - about 156
- UCM Data Objects layer**
 - about 30
- UCM Dispatcher**
 - about and user properties 62
 - arguments 42
 - mapping integration objects, about 59
 - methods 41
 - user properties 41
 - viewing map integration objects user properties 59
- UCM envelope integration objects, modifying** 58
- UCM FINCORP Account messages** 113
- UCM Household messages**
 - about 107
 - Household Delete Integration Components 108
 - Household Insert Integration Components 107
 - Household Integration Component Fields 108
 - Household Query by Name Integration Components 107
 - Household Query Integration Components 107
 - Household Update Integration Components 107
- UCM Personal Address messages**
 - about 112

- FINCORP Account Integration Component
 - Fields 113
 - FINCORP Account Query Integration
 - Components 113
 - Personal Address Integration Component
 - Fields 112
 - Personal Address Query Integration
 - Components 112
 - UCM FINCORP 113
 - UCM Publish/Subscribe Service**
 - argument details for publish method (Daily Batch) 45
 - argument details for publish method (Real Time) 45
 - method arguments 45
 - methods 44
 - UCM Security Manager**
 - method arguments 43
 - method Security Check 44
 - methods 43
 - user properties 42
 - UCM Security Manager, about and user properties** 62
 - UCM Transaction Manager**
 - about and user properties 31
 - execute method arguments 34
 - method arguments 33
 - Operation user properties 60
 - pre-built methods 32
 - UCM user interface layer** 22
 - universal identification**
 - secure repository for customers' social security numbers 10
 - universally unique identifier, about** 76
 - Update functionality** 18
 - user interface layer** 22
 - UUID (universally unique identifier), about** 76
- W**
- workflow integration**
 - support of Siebel Industry eAI architecture 25
- X**
- XML-based applications, about integrating** 20
- Z**
- zero downtime, designed for** 21