Siebel Master Data Applications Reference for Industry Applications

Universal Customer Master Guide

Version 7.8, Rev. B

April 2006

ORACLE | SIEBEL

Contents

Chapter 6: Configuring Siebel Master Data Applications

Chapter 7: Administering Siebel Master Data Applications

Chapter 8: Siebel Master Data Applications Messages

Appendix A: Additional License Restrictions

Index

1 What's New in This Release

What's New in Siebel Master Data Applications Reference for Industry Applications, Version 7.8, Rev. B

Table 1 lists changes described in this version of the documentation to support Release 7.8.2 of the software.

Table 1. What's New in Siebel Master Data Applications Reference for Industry Applications, Version 7.8, Rev. B

Topic	Description
"About Siebel Universal Customer Master Concepts" on page 15	Added this topic to give a further overview of features and functionality included with Siebel Universal Customer Master.
"About Siebel Master Data Application Workflows" on page 34	Added information on new workflows
"About the Universally Unique Identifier (UUID)" on page 37	Consolidated information on UUID in this location.
Chapter 5, "Installing Siebel Master Data Applications"	Added this new chapter on Siebel UCM installation details.
Chapter 6, "Configuring Siebel Master Data Applications"	Extensively reorganized this chapter to combine like configurations.
"Configuring Initial Setup for Siebel Master Data Applications" on page 45	Added new topics on initial configurations to Siebel UCM required after installation.
"Configuring Siebel Master Data Application Features" on page 46	Consolidated existing application configurations in this new topic.
"Configuring Siebel Master Data Integration Objects" on page 67	Consolidate existing integration configurations in this new topic.
"About UCM Data Management Services" on page 56	Added new conceptual topic on UCM data management services.
"Configuring Deep Copy for Data Hierarchies in UCM" on page 65	Added conceptual information on UCM and CRM integration.
"About System Publish and Subscribe Modes" on page 78	Added conceptual information on UCM Publish and Subscribe modes.
"About System Preferences for Siebel UCM" on page 96	Added information and brief descriptions of UCM system preferences.

Table 1. What's New in Siebel Master Data Applications Reference for Industry Applications, Version 7.8, Rev. B

Topic	Description
"Configuring Siebel Data Quality Cleansing for Siebel UCM" on page 97	Add topic on configuring data quality cleansing.
"Configuring SSA Data Matching for Siebel UCM" on page 99	Added topic on configuring data matching.
"Troubleshooting Siebel Master Data Applications" on page 102	Added a list of 7.8.2 specific error messages with resolutions.
"About CRMML Messages" on page 116	Consolidated information on CRMML messages in this topic and chapter.
"Configuring UCM Data Quality Manager for Exact Match" on page 57	Added this topic on configuring exact match.
"Configuring Run-time Events for UI Interaction" on page 60	Added this topic on configuring run-time events for UI interaction.
"Configuring Default System for UI Interaction" on page 61	Added this topic on configuring the default system for UI interaction.
"Configuring Siebel UCM List of Values" on page 65	Added this topic on configuring UCM List of Values.

What's New in Siebel Master Data Applications Reference for Industry Applications, Version 7.8, Rev. A

Table 2 lists changes described in this version of the documentation to support Release 7.8 of the software.

Table 2. What's New in Siebel Master Data Applications Reference for Industry Applications, Version 7.8, Rev. A

Topic	Description
"Generation of Party_UID" on page 34	Added this topic on the generation of the Party_UID field.
"About CRMML Messages" on page 116	Updated this topic with information on CRMML customization.

What's New in Siebel Master Data Applications Reference for Industry Applications, Version 7.8

Table 3 lists changes described in this version of the documentation to support Release 7.8 of the software.

Table 3. What's New in Siebel Master Data Applications Reference for Industry Applications, Version 7.8

Topic	Description
"About Integration Services for Siebel Master Data Applications" on page 24	Added further background capabilities based on new 7.8 integration services.
"Siebel Connector and Integration of Siebel Master Data Applications" on page 30	Listed and added details on the new 7.8 business services.
"About Siebel Master Data Application Workflows" on page 34	Added information on Siebel Master Data Application workflows.
"About Siebel Universal Customer Master Server Components" on page 37	Added information on new Siebel Universal Customer Master server components.
Chapter 7, "Administering Siebel Master Data Applications"	Added new administrative tasks for the following 7.8 features: survivorship rules and data management.
Chapter 6, "Configuring Siebel Master Data Applications"	Created a new configuration chapter that contains tasks to configure 7.8 functionality, in addition to configuration tasks that were previously documented in the former administrative chapter.

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 12.

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 Business 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.

For information on integrating Siebel Master Data Applications, see Chapter 6, "Configuring Siebel Master Data Applications."

The chapter includes the following topics:

About Siebel Master Data Applications Modules

Siebel Master Data Applications include the base module (Siebel Universal Customer Master) and the following 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 13
- "Siebel Universal Sales Master" on page 13
- "Siebel Universal Service Master" on page 13
- "Siebel Universal Field Service Master" on page 13
- "Siebel Universal Identification Master" on page 14

Siebel Universal Customer Master

The Siebel Universal Customer Master is a Siebel platform configured to store a clean 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.

For further information on concepts and functionality of the Siebel Universal Customer Master, see "About Siebel Universal Customer Master Concepts" on page 15.

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 Business 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 Business 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 Business Application Architecture. See *Siebel System Administration Guide* for further information.

- Siebel Enterprise Application Integration (Siebel EAI)

  Siebel EAI provides components for integrating Siebel Business Applications with external applications and technologies. See *Overview: Siebel Enterprise Application Integration* 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 Business 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 Business Applications, including Data Layer, Object Layer, User Interface Layer, and publish/subscribe services. See *Configuring Siebel Business Applications* 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 Business 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 Business Applications to access and use D&B data and reports. See *Applications Administration Guide*.

About Siebel Universal Customer Master Concepts

Siebel Universal Customer Master (Siebel UCM) is a Siebel platform configured to store a clean and unified profile for enterprise customer data. Siebel UCM is the primary Master Data Applications product and serves as the example for installation, configuration, and administration tasks that are further documented in the following chapters.

In addition to storing the master set of customer data for an enterprise, Siebel UCM also includes many features to cleanse, evaluate, publish, store, and manage this customer data. The basic Siebel UCM concepts and functionality are briefly described in Table 4.

Table 4. Siebel UCM Features

Feature	Description
UCM Survivorship	The UCM Survivorship feature provides a rules-based means to automate the quality of the master customer data. Data is compared to its source and age to determine whether to maintain or update customer data. For further information on UCM Survivorship, see "About UCM Survivorship Rules" on page 82.
UCM Cross-referencing	UCM cross-referencing allows the identification of customer data in external systems to be saved in the Siebel UCM. For further information on UCM cross-referencing, see "About Cross-Referencing Records with External Systems" on page 90.
UCM Best Version Records	UCM Best Version records describe the current best state of the customer data stored in the Siebel UCM. These records may be updated in the future based on survivorship rules or other data management processes. Historical best versions of customer data are also stored in the UCM Source Data History table.
UCM Source Data History (SDH) Table	The UCM Source Data History (SDH) tables maintain a record of data transactions between Siebel UCM and registered external systems. For further information on SDH tables, see "About SDH Tables" on page 91.
UCM Publish and Subscribe	UCM publish and subscribe functionality determines the details on how external systems receive customer data updates from the Siebel UCM. For further information on UCM publish and subscribe, see "About System Publish and Subscribe Modes" on page 78 and "Configuring System Publish and Subscribe for Siebel Master Data Applications" on page 79.
UCM Data Cleansing and Matching	Siebel UCM supports data cleansing using the Siebel Data Quality module and data matching using SSA-NAME3 (SSA) third-party software. Siebel UCM also supports other third-party cleansing and matching technologies. For further information on configuring UCM data cleansing and matching, see "Configuring Siebel Data Quality Cleansing for Siebel UCM" on page 97 and "Configuring SSA Data Matching for Siebel UCM" on page 99.

Table 4. Siebel UCM Features

Feature	Description
Siebel Connector for Master Data Applications	Siebel Connector for Master Data applications describes a set of preconfigured business services used to administer Siebel UCM processes. For further information on the Siebel Connector for Master Data Applications, see "Siebel Connector and Integration of Siebel Master Data Applications" on page 30.
UCM CRMML Messages	Siebel UCM CRMML messages are a Siebel-specific XML messaging format used for UCM data exchange. For further information on CRMML messages, see "About CRMML Messages" on page 116.

3 Siebel Master Data Applications Scenario

This chapter examines a fictional financial institution, its business requirements, and the Siebel Master Data Applications product solution based on the following scenario: a stand-alone UCM deployment. The UCM could also be deployed on the same CRM instance.

For background information about Siebel Master Data Applications, see Chapter 2, “About Siebel Master Data Applications.”

Siebel Universal Customer Master Scenario

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 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.

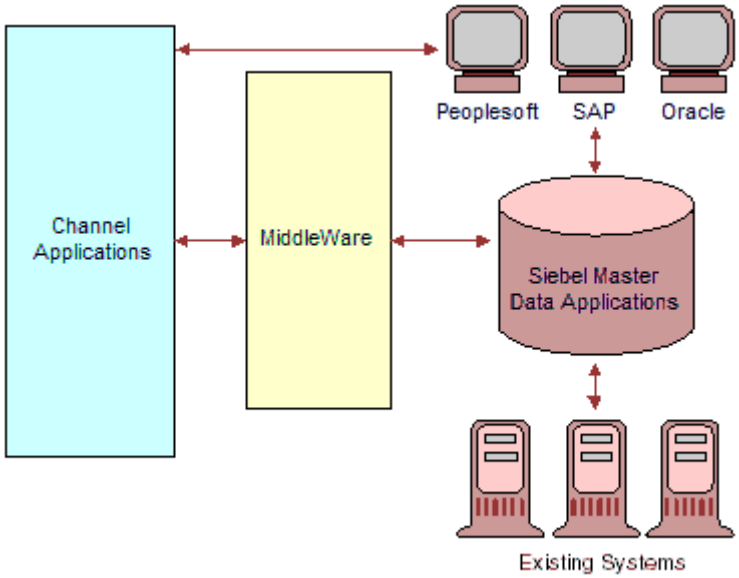

Figure 1. Siebel Master Data Applications Stand-Alone Deployment Scenario

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 30 for details about this process. This scenario can also be deployed within a Universal Application Network (UAN) framework.

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.

The chapter includes the following topics:

About Siebel Master Data Applications Framework and Interactions

Siebel Master Data Applications form the master application and database of an organization's data. It interacts 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 Business Applications or back-office applications using enterprise integration and the regular links to Siebel Business Applications. See Figure 2 for this relationship.

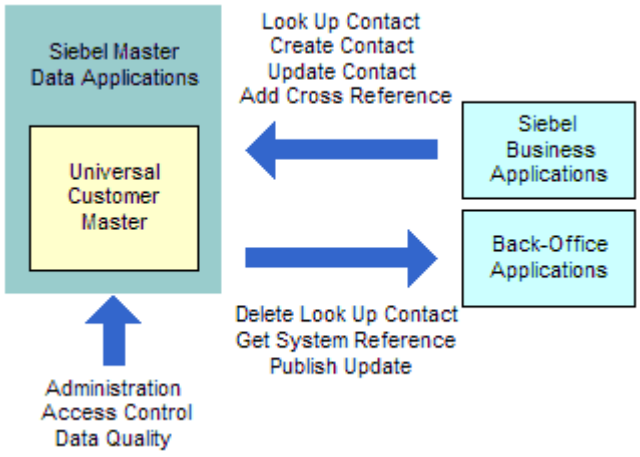

Figure 2. Siebel Master Data Applications Framework

- For information about the functionality included with Siebel Master Data Applications, see "About Siebel Master Data Applications Functions and Services" on page 22.
- For information about the administrative functionality of Siebel Master Data Applications, see "Administering Siebel Master Data Applications" on page 23.
- 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 30.

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 on master data, such as customers and accounts. When these functions are available in Siebel Business 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. For further information on Siebel Connector for Master Data Applications, see "About Integration Services for Siebel Master Data Applications" on page 24 and "Siebel Connector and Integration of Siebel Master Data Applications" on page 30.

For details about prebuilt Siebel Master Data Application services for accessing these functions, see Chapter 8, "Siebel Master Data Applications Messages."

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

- **Publish.** Publish data records from Siebel Master Data Applications to subscribing applications. For further information on administering this functionality, see "Setting System Privileges for Siebel Master Data Applications" on page 76.
- **Request and Reply.** Other applications send a request for data from Siebel Master Data Applications. For further information on configuring request and reply, see "Setting System Privileges for Siebel Master Data Applications" on page 76.

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 25 for further information.

Administering Siebel Master Data Applications

Siebel Master Data Applications are administered through Siebel Customer Data Steward user interface screens. However, only those screens necessary for viewing and administering the information contained within Siebel Master Data Applications are included. See Chapter 7, "Administering Siebel Master Data Applications" for further information on administration. Only administrative users have access to these screens. An example of one of the administration screens appears in Figure 3.

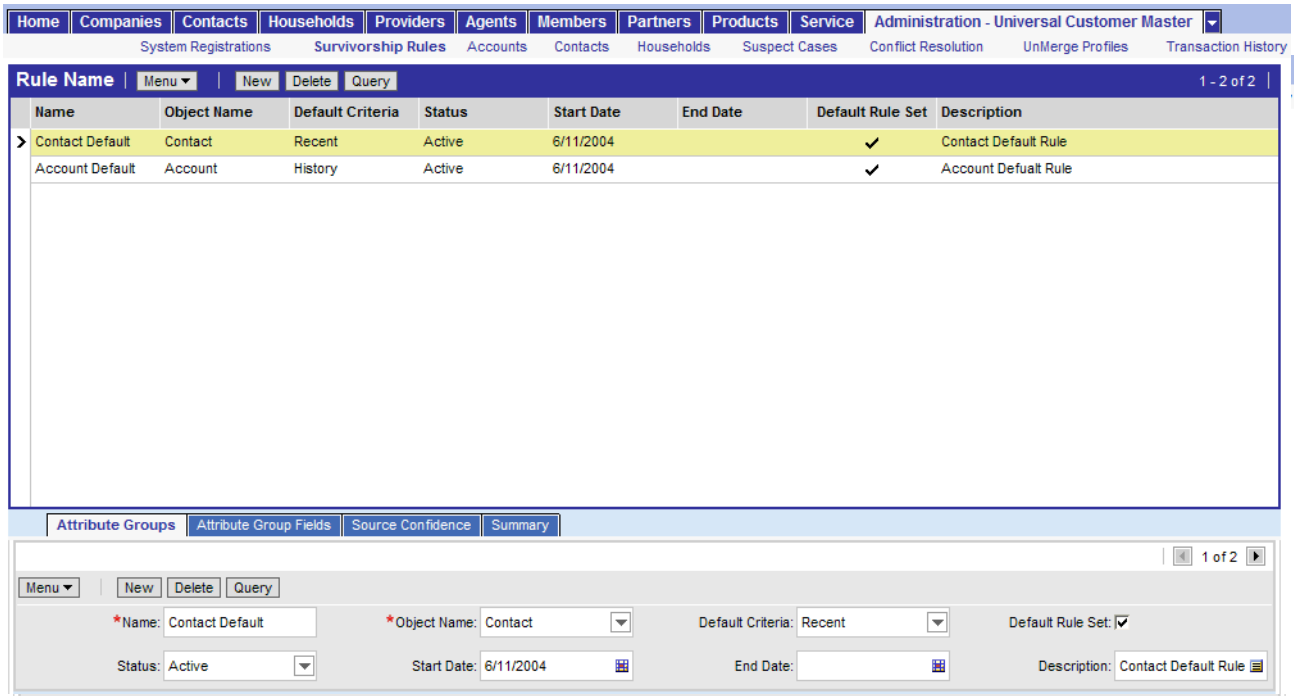

Figure 3. Siebel Business Applications Administration - Universal Customer Master screen

The administration screens are used to make manual changes to the data, make decisions on suspect information, and set-up rules and policies for UCM data management.

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 on configuring system privileges, see "Setting System Privileges for Siebel Master Data Applications" on page 76.

For general information and procedures on access control, see *Configuring Siebel Business Applications* and *Security Guide for Siebel Business Applications*.

About Integration Services for Siebel Master Data Applications

The Siebel Connector for Master Data Applications provides the integration services for Siebel Master Data Applications, and is a configurable set of components, which you can use to exchange data between Siebel Master Data 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 Master Data 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 Scenario" for scenarios using the Siebel Master Data Applications.

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

- Registration 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 "Cross-Referencing Records with External Systems" on page 90.
- Running of intelligent merge and update
- Running of point-in-time recovery, including unmerge functionality
- Batch data management
- Enabling survivorship rules in UCM

Within the Siebel Master Data Applications, the Siebel Connector offers the following:

- Generation of Universal Unique IDs (UUID)
- Creation of source data and history records

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

About Siebel Master Data Applications Architecture

Siebel Master Data Applications are based on the Siebel Business Application n-tier object architecture in which the user interface, business logic, and data are separated and layered, see Figure 4. 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.

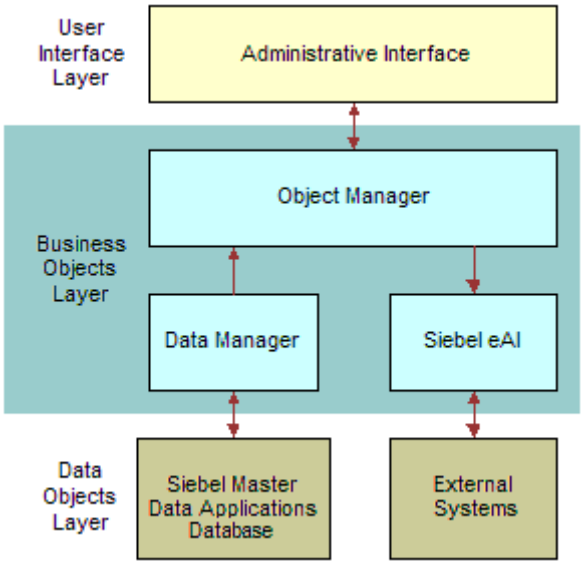

Figure 4. Siebel Master Data Applications n-tier Architecture

The Siebel Master Data Applications platform is designed for 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 Business Applications*. The user interface layer is used for the administration of the Siebel Business Information suite; see "Administering Siebel Master Data Applications" on page 23 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 Business 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 Business Applications*.

Siebel Industry EAI Architecture

Siebel Industry EAI architecture is built to work with the Siebel Enterprise 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 Business 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 5 illustrates the high-level architecture of the Siebel Industry EAI and the standard connectors.

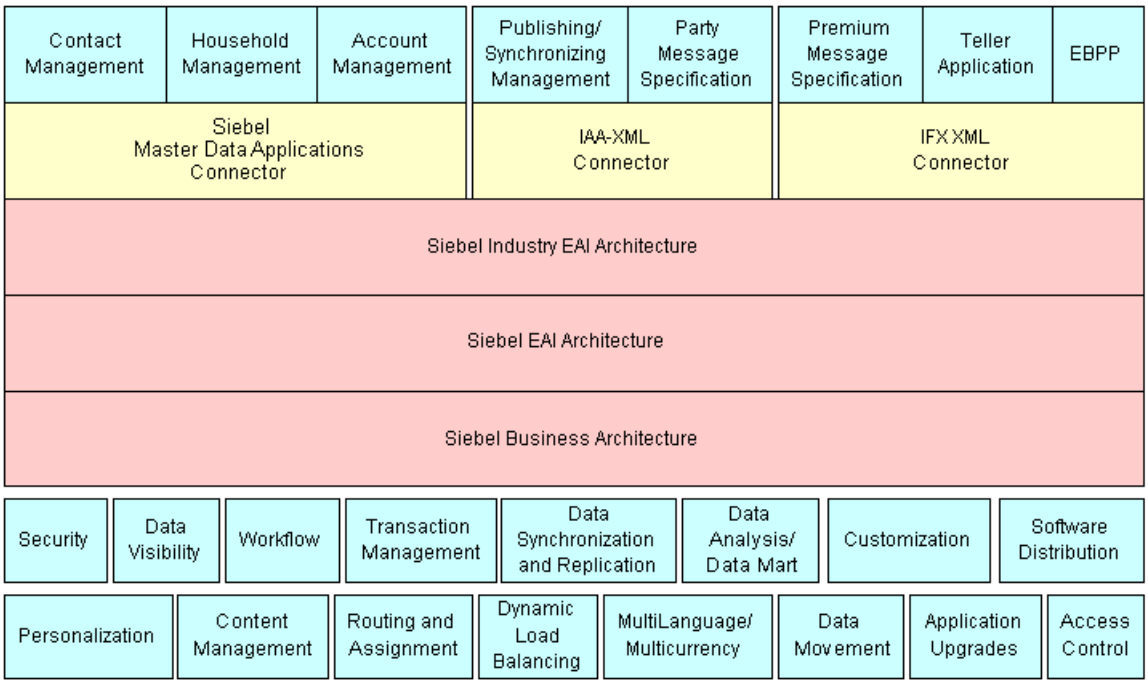

Figure 5. High-Level Architecture of Siebel Industry EAI and Standard Connectors

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 several business services. For further information on these services, see "Siebel Connector and Integration of Siebel Master Data Applications" on page 30.

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 6 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.

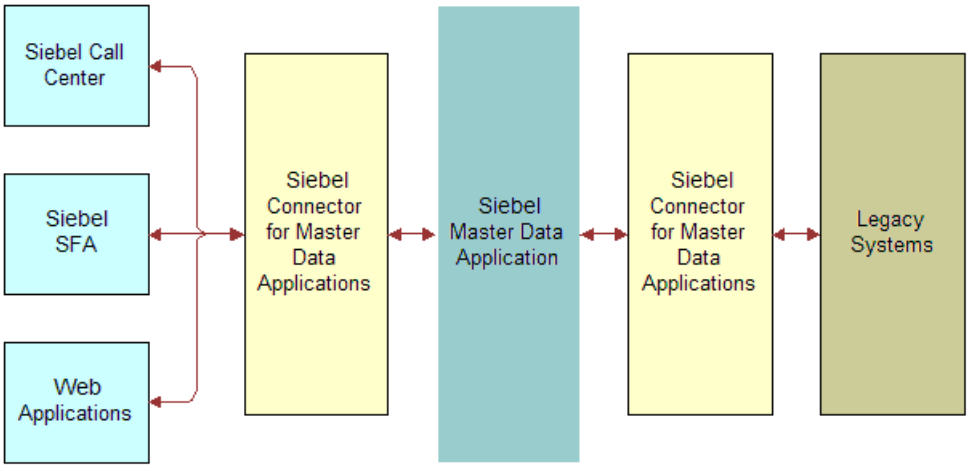

Figure 6. Business Process Flow

Inbound Data Flow

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 68.

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. Source data is captured for insert, update, and delete operations, and history data is captured for update and delete operations.
   - Insert requests are queued in source data history table.
   - Update operations invoke UCM Survivorship Engine or UCM Data Quality Manager, if enabled.
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 Business Applications. You create the workflow processes using a graphical user interface provided within Siebel Business 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 Enterprise Application Integration*.

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, see *Integration Platform Technologies: Siebel Enterprise Application Integration*.

CRMML Message

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.

For further information on CRMML, see "About CRMML Messages" on page 116.

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 run-time 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 Business 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 the following prebuilt business services:

- UCM Transaction Manager
- UCM Converter
- UCM Dispatcher
- UCM Security Manager
- UCM Publish/Subscribe Service
- UCM Batch Manager
- UCM Data Quality Manager
- UCM Survivorship Engine
- UCM Account Source Data and History Service
- UCM Contact Source Data and History Service
- UCM UI Interaction Service

Use Siebel Tools to review specific details on business services, including user properties, methods, and method arguments. For information on using Siebel Tools, see *Using Siebel Tools*. See also "About Siebel Master Data Applications Integration Services" on page 47.

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.

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.

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.

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 Universal 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.

UCM Publish/Subscribe Service

The UCM Publish/Subscribe Service supports both near real-time and daily-batch publish types.

The near 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 near 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.

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

UCM Batch Manager

UCM Batch Manager processes insert requests queued in the source data history table in batch mode.

UCM Data Quality Manager

UCM Data Quality Manager is responsible for cleansing and matching records before processing a transaction.

UCM Survivorship Engine

UCM Survivorship Engine is a prebuilt business service to execute survivorship logic defined by data administrators. For background information on administering survivorship, see "About UCM Survivorship Rules" on page 82.

UCM Account Source Data and History Service

UCM Account Source Data and History service synchronizes the Account record with the UCM Account Source Data and History record.

UCM Contact Source Data and History Service

UCM Contact Source Data and History service synchronizes the Contact record with the UCM Contact Source Data and History record

UCM UI Interaction Service

UCM UI Interaction Service captures attribute group data for survivorship during the user interface insert or upsert. It also captures the source data history records during the update or delete process.

Transport Adapter

Transport Adapter is a prebuilt business service providing an interface between Siebel Business 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 Enterprise Application Integration*.

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. Unlike Siebel EAI, Siebel EIM does not go through a application object manager, but the data moves from the EIM_UCM_ORG interface table directly into the SDH (S_UCM_ORG) tables. Siebel EIM also 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 batch integration with external applications, Siebel UCM provides the following UCM-specific EIM tables to facilitate the loading of selected master objects into UCM's Source Data and History tables. These records can then be processed by UCM data management tasks at a later date:

- EIM_UCM_ORG
- EIM_UCM_CON
- EIM_UCM_CON_CHLD
- EIM_UCM_ORG_CHLD

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

Generation of Party_UID

Siebel EIM and Siebel Master Data Applications workflow processes generate the Party_UID field based on the value in the UCM_EXT_ID field. The process for generating this field occurs by either of the following means:

- **Loading data to UCM SDH table using UCM Server Party Package Workflow**. In this scenario, the user must populate the <ID> integration object component field value in the CRMML message, which maps to UCM_EXT_ID field in the UCM SDH table. Additionally, the <PartyUId> field, which maps to PARTY_UID, must be empty or not sent with the CRMML message. The UCM Server Party Package workflow then generates the PARTY_UID based on UCM_EXT_ID.
- **Loading data to UCM SDH table using Siebel EIM**. In this scenario, users must populate the UCM_EXT_ID column value in the EIM interface table and the UCM_PARTY_UID field value must remain empty. The EIM interface table value populates the UCM_EXT_ID field in the SDH table. The UCM Batch Data Management workflow then generates the PARTY_UID based on the UCM_EXT_ID.

For further information on the Party_UID, see *Siebel Enterprise Integration Manager Administration Guide*.

About Siebel Master Data Application Workflows

Workflows are a feature of Siebel Master Data Applications. They automate the business processes associated with managing data stored in the master data application.

You can modify these workflows to suit your own business model using Siebel Business Process Designer. For more information, see *Siebel Business Process Designer Administration Guide*.

To view the diagrams of the following workflows, use the Workflow Process Designer view in Administration - Business Process screen.

The following workflows describe the business processes for the Siebel Universal Customer Master (Siebel UCM):

- “UCM Batch Process - Single Step” on page 35

UCM Batch Process - Single Step

This workflow mainly calls the UCM Batch Manager business service to process the queued Insert requests or records imported through EIM. The workflow looks for queued or batch records in the Source Data History table, then calls UCM Data Quality Manager to cleanse and match the records. Depending on the match results, it then calls Transaction Manager to execute the insert or update operation, or store the cleansed version for later deduplication processing.

NOTE: This workflow may be configured to be invoked by the UCM Batch Process server component (alias UCMBatchProcess).

UCM Server Party Package Workflow (MQSeries Server Transport)

This workflow handles incoming requests. It converts and dispatches the incoming request message, checks if the source system has privileges for the requested operations, executes the operations, captures source version, captures History version as necessary, invokes CDM and Survivorship as necessary, and publishes to other subscribing systems. It is expected that this workflow process is the dispatch workflow for an MQSeries Receiver.

This workflow receives data as a CRMML message, which differs the UCM Batch Process workflow in that it does not going through the SDH tables.

UCM Server Party Package Workflow (MQSeries Server Transport Error Handling)

This workflow is the same as "UCM Server Party Package Workflow (MQSeries Server Transport)" on page 35 but additionally handles errors.

This workflow receives data as a CRMML message, which differs from the UCM Batch Process workflow in that it does not go through the SDH tables.

NOTE: As this workflow reads in the input from file, it is intended for debugging purposes.

UCM Server Party Package Workflow (File Transport)

This workflow is the same as "UCM Server Party Package Workflow (MQSeries Server Transport)" on page 35 but additionally handles reading the request message from a file and writing the response message to a file.

UCM Async Real-time Publish Workflow

This workflow is invoked as part of a server request to perform asynchronous real-time publishing. It is called from either the connector workflow (for example, UCM Server Party Package Workflow (MQSeries Server Transport)) or Batch Data Management workflow (UCM Batch Process) when the real-time Publish/Subscribe flag is true.

In this workflow, the second step, RealTimePublish, calls the UCM Publish/Subscribe service with the method RealTimePublishMethod(). This method of UCM Publish/Subscribe Business Service handles the actual publishing of the message to the registered systems.

Real-time, asynchronous, publishing increases the performance for both the Connector workflow and the batch data management workflow.

UCM Daily Publish Workflow

This workflow is a one-step workflow calling the PublishMethod without sending in any input message. When run from UCM Batch Publish Subscribe (alias UCMBatchPubSub) server component, the Workflow Process Manager (alias WfProcMgr) triggers this workflow.

There are three server component parameters that UCMBatchPubSub reads during this process:

- UCM Batch Object Type (alias UCMBatchObjectType)
- UCM Batch Size (alias UCMBatchSize)
- UCM Sleep Time (alias UCMSleepTime)

This workflow publishes all the new and updated records to the registered systems with Publish Frequency set as Daily Batch.

UCM Account Batch Data Management Workflow

This workflow performs the same task as the UCM Batch Process - Single Step workflow but only processes Account records. The steps are exposed so they can be configured if necessary. The Workflow Process Batch Manager server component (alias WfProcBatchMgr) retrieves the IDs of records to be processed and then invokes this workflow to cleanse and match account data.

UCM Contact Batch Data management Workflow

This workflow performs the same task as the UCM Batch Process - Single Step workflow but only processes Contact records. The steps are exposed so they can be configured if necessary. The Workflow Process Batch Manager server component (alias WfProcBatchMgr) retrieves the IDs of records to be processed and then invokes this workflow to cleanse and match account data.

UCM Batch Process Workflow

This workflow also performs the same task as the UCM Batch Process - Single Step workflow, but the steps process multiple records instead of one record at a time. It will be invoked by the UCM Batch Process server component (alias UCMBatchProcess).

About Siebel Universal Customer Master Server Components

The Siebel Universal Customer Master (UCM) has one server component group and three server components to administer the functionality of the Siebel UCM. The component group's name is Siebel Universal Customer Master (alias UCM). The UCM component group contains the following three server components:

Server Component	Alias	Description
UCM Object Manager	UCMObjMgr	An interactive-mode server component, the UCM Object Manager is the application object manager that administers all requests and data management tasks to the UCM.
UCM Batch Manager	UCMBatchProcess	A batch-mode server component that manages batch invocation of Customer Data Management for account and contact insert requests that have been queued in source data history table; this server component utilizes the UCM Batch Process workflow.
UCM Batch Publish Subscribe	UCMBatchPubSub	A batch-mode server component that manages the batch publishing of new and updated records to subscribing external systems.

For background information on configuring and managing server components, see *Siebel System Administration Guide*.

About the Universally Unique Identifier (UUID)

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

5 Installing Siebel Master Data Applications

This chapter contains information for installing Siebel Master Data Applications. The following topics are included:

Process of Installing Siebel Master Data Applications

The Siebel Server installation and configuration process requires multiple tasks that you should perform in the following sequence:

1. Install and configure the Siebel Server. For more information, see "Installing and Configuring the Siebel Server" chapter in the Siebel Installation Guide for the operating system you are using.
2. Install Siebel Master Data Applications. See "Installing Siebel Master Data Applications" on page 39.
3. Import Siebel Master Data Applications import files (SIFs). See "Importing Siebel Master Data Applications Import Files" on page 41.
4. Import Siebel Master Data Applications workflows. See "Importing Siebel Master Data Applications Workflows" on page 42.
5. Complete Siebel Master Data Applications post-installation tasks. See "Completing Siebel Master Data Applications Post-Installation Tasks" on page 43.

Installing Siebel Master Data Applications

The following instructions use the Siebel Universal Customer Master (Siebel UCM) as the example that covers all of the Siebel Master Data Applications.

Siebel UCM is enabled in the Siebel application by using a license key, which is received in a separate letter from Siebel Systems. This supplementary media contains objects and data for you to import into your Siebel environment.

To install the Siebel UCM release

1 In Windows Explorer, navigate to the Siebel image location for the current software version. Then navigate to the directory where the installer is located.

In this case, navigate to *Siebel_Image**Windows**Server**Siebel_UCM*

where:

Siebel_Image = The directory for your version-specific Siebel network image, such as D:\Siebel_Install_Image\7.8.2.0.

- Double-click setup.exe.

The Welcome to InstallShield Wizard for Siebel Universal Customer Master screen appears.

Go to Step 3 on page 40.

2 For the UNIX operating system:

a Open a new shell and navigate to the Siebel image location for the current software version. Then navigate to the directory where the installer is located.

For example, for the AIX operating system, navigate to *Siebel_Image*/*AIX*/Server/Siebel_Enterprise_Server.

where:

- *Siebel_Image* = The directory for your version-specific Siebel network image, such as /Siebel_Install_Image/7.8.2.0.

b Unset any Siebel-related environment variables, such as $*SIEBEL_ROOT*.

c Enter the following command. Optionally, append any flags described in "Additional Flags for Installation Commands" in the *Siebel Installation Guide for UNIX: Servers, Mobile Web Clients, Tools*.

./setup*aix*

where:

- *Siebel_Image* = The directory for your version-specific Siebel network image, such as /export/home/Siebel_Install_Image/7.8.2.0.

For the AIX operating system, execute ./setupaix.

The Welcome to InstallShield Wizard for Siebel Universal Customer Master screen appears.

Go to Step 3 on page 40.

3 Click Next.

4 Use the Browse button to select a different drive or directory or select the displayed default directory for file installation, and then click Next.

This directory name must not contain spaces, although underscores are allowed.

5 The installer program performs a validation check to make sure that installation prerequisites were met. If they were not, it errors out.

The installer displays the location into which it will install the Siebel Universal Customer Master. It also displays the disk space required for the software you are installing.

Click Next.

6 Review the information and take the appropriate action:

- If these parameters are acceptable, click Next.
- Otherwise, click Back to adjust your installation parameters or location.

The installer proceeds to install the specified files.

7 Review the information on the Siebel Universal Customer Master Installation Instructions screen.

This information may contain more installation and configuration instructions.

Click Next.

8 Click Finish to exit the InstallShield wizard.

Importing Siebel Master Data Applications Import Files

Following the installation of Siebel Master Data Applications, import Siebel Import Files (SIFs) into your development repository using Siebel Tools. Siebel Universal Customer Master (Siebel UCM) sif files are available in the package UCMArchives.zip.

This task is a part of the "Process of Installing Siebel Master Data Applications" on page 39.

For further information on using Siebel Tools, see *Using Siebel Tools*.

CAUTION: If you are importing into your existing development repository, which has modified objects belonging to the applicable projects, then:

- Export any changes made into SIF files.
- Follow the procedure below to import changes and make sure to check the Overwrite the object definition in the repository radio button in the Preview screen.
- Compile all the changes, and test to make sure that preconfigured Siebel UCM functions are working properly.
- Then import all the exported development changes from your SIF file to continue development.

To import Siebel UCM SIF files

1 Unzip the UCMArchives.zip file and save the .sif files to your computer. The UCMArchives.zip contains the following SIF files:

Account BC.sif	UCM Merge Project.sif

Account BO.sif	UCM SDH Project.sif
Contact BC.sif	UCM Survivorship Proejct.sif
Contact BO.sif	UCM Unmerge Project.sif
UCM CDM Child Project.sif	VEAI UCM Account ASI Project.sif
UCM Data Management Project.sif	VEAI UCM CIF Project.sif
UCM Hierarchy Management Project.sif	VEAI UCM CIF UI Project.sif

2. Back-up a copy of your existing Tools repository using the repimexp utility. For further information on this task, see *Going Live with Siebel Business Applications*.
3. Launch Siebel Tools and connect to your development server.
4. Navigate to Project in the Object Explorer window; query for the following projects and lock them: Account, Contact, UCM CDM Child, UCM Data Management, UCM Merge, UCM SDH, UCM Survivorship, UCM Unmerge, UCM Hierarchy Management, VEAI UCM Account ASI, VEAI UCM CIF, and VEAI UCM CIF UI.
5. Select Tools > Import From Archive.
6. In the Select Archive to Import box, navigate to the location of the *.sif file and click Open.
7. In the Import Wizard - Preview box, click the Overwrite the object definition in the repository radio button and click Next.
8. After Siebel Tools has read the file, click Next.

   A dialog box appears stating number of objects that will be modified in your repository. Click Yes.
9. In the Import Wizard - Summary box, click Finish.
10. Repeat steps 5) through 10) for each .sif file.
11. Compile all affected projects into a new Siebel Repository File (.srf).

Importing Siebel Master Data Applications Workflows

Following the installation of Siebel Import Files (SIFs), import Siebel Master Data Applications into your development repository using Siebel Tools. Siebel Master Data Application workflows contain the business process flows for Siebel Universal Customer Master (Siebel UCM).

Siebel UCM workflows are available in the package UCMWorkflows.zip.

This task is a part of the "Process of Installing Siebel Master Data Applications" on page 39.

For further information on using Siebel Tools, see *Using Siebel Tools*.

To import Siebel UCM workflows

1. Unzip the UCMWorkflows.zip file and save the XML workflow files to your computer. The UCMWorkflows.zip contains the following Siebel UCM workflows:

   UCM Account Batch Data Management Workflow.xml
   UCM Async Realtime Publish Workflow.xml
   UCM Batch Process - Single step.xml
   UCM Batch Process.xml
   UCM Contact Batch Data Management Workflow.xml
   UCM Daily Publish.xml
   UCM Server Party Package Workflow (File Transport).xml
   UCM Server Party Package Workflow (MQSeries Server Transport Error Handling).xml
   UCM Server Party Package Workflow (MQSeries Server Transport).xml

2. Launch Siebel Tools and connect to your Siebel database.
3. Navigate to Project in the Object Explorer window, and query for the Siebel Workflows - Seed project and lock it.
4. Navigate to Workflow Process in the Object Explorer window.
5. Right-click on the Workflow Process applet and select Import Workflow Process from the applet menu.
6. In the Workflow Process Import box, navigate to the location of the *.xml file and click Open.
7. In the Select Project box, select the project name and click OK.
8. Repeat steps 5) to 7) for each Workflow *.xml file.

Completing Siebel Master Data Applications Post-Installation Tasks

After completing the installation process, you must make initial configurations to prepare the Siebel Master Data application, as well as configure functionality for your business needs. For information on initial configuration tasks, see "Configuring Initial Setup for Siebel Master Data Applications" on page 45. For other configuration tasks, see Chapter 6, "Configuring Siebel Master Data Applications."

Optionally, if you are configuring integration with Siebel UCM and a CRM instance, review the configuration information in the `crm782-ucm782-intgn-readme.txt` included with the installation.

6 Configuring Siebel Master Data Applications

This chapter describes background information and configuration procedures for Siebel Master Data Applications.

The following topics describe procedures for configuring the Siebel application with Siebel Tools. Review background information on configuring Siebel applications, using Siebel Tools, and configuring integration objects in the following documentation:

- *Configuring Siebel Business Applications*
- *Using Siebel Tools*
- *Integration Platform Technologies: Siebel Enterprise Application Integration*

This chapter includes configuration topics in the following subareas:

- "Configuring Initial Setup for Siebel Master Data Applications" on page 45
- "Configuring Siebel Master Data Application Features" on page 46
- "Configuring Siebel Master Data Integration Objects" on page 67

Configuring Initial Setup for Siebel Master Data Applications

After installation of Siebel Master Data Applications (Siebel MDA), perform the following bullet-list tasks to configure the master data application—and any external applications accessing the Siebel MDA—at the initial setup phase.

- "Loading Data into Siebel UCM and Updating External Systems" on page 46
- "Activating Siebel UCM External Systems" on page 46

See the following related topics for further information on other areas of application configuration and the installation of Siebel MDA.

Related Topics

"Configuring Siebel Master Data Application Features" on page 46

"Configuring Siebel Master Data Integration Objects" on page 67

"Installing Siebel Master Data Applications" on page 39

Loading Data into Siebel UCM and Updating External Systems

After installation of Siebel Universal Customer Master (Siebel UCM), the initial customer data must be loaded into the database. In preparation for the this process, it is recommended to:

- Cleanse the data in each external system prior to loading data
- Load the best or highest quality data first

Use Siebel Enterprise Integration Manager (Siebel EIM) to load bulk data records into the Siebel UCM. For further information on Siebel EIM, see "Siebel EIM and Integration of Siebel Master Data Applications" on page 33 and *Siebel Enterprise Integration Manager Administration Guide*.

Activating Siebel UCM External Systems

After loading the customer data into the Siebel Universal Customer Master (Siebel UCM), you must register and activate external systems that access the customer master data.

For information on these procedures, see:

- "Registering Systems Connected to Siebel Master Data Applications" on page 73
- "Setting System Privileges for Siebel Master Data Applications" on page 76

Additionally, other Siebel UCM application features may require configuration or activation.

For further information, see:

- "Configuring Siebel Master Data Application Features" on page 46
- Chapter 7, "Administering Siebel Master Data Applications"

To further customize the Siebel MDA, see "Configuring Siebel Master Data Integration Objects" on page 67.

Configuring Siebel Master Data Application Features

This topic contains a list of general application feature configurations you can perform after installing and initially configuring Siebel Master Data Applications.

- "Configuring UCM Unmerge Functionality" on page 58
- "Configuring UCM Server Components for Batch Transactions" on page 59
- "Configuring Run-time Events for UI Interaction" on page 60
- "Configuring Default System for UI Interaction" on page 61
- "Configuring UCM Inbound Server Communication" on page 61
- "Configuring Siebel UCM List of Values" on page 65
- "Configuring Deep Copy for Data Hierarchies in UCM" on page 65
- "Configuring Deep Copy for Data Hierarchies in UCM" on page 65

See the following related topics for further information on other areas of application configuration and the installation of Siebel MDA.

Related Topics

"Installing Siebel Master Data Applications" on page 39

"Configuring Initial Setup for Siebel Master Data Applications" on page 45

"Configuring Siebel Master Data Integration Objects" on page 67

About Siebel Master Data Applications Integration Services

Siebel Master Data Applications (Siebel MDA) provide an object called a business service, which you can reuse in multiple applications. These business services together perform the insert, update, query, and delete operations on Siebel MDA data. You can use these business services together in the prebuilt workflow processes or build your own workflow processes or business services.

The Siebel Connector for the Master Data Applications 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
- UCM Batch Manager
- UCM Data Quality Manager
- UCM Survivorship Engine
- UCM Account Source Data and History Service
- UCM Contact Source Data and History Service
- UCM UI Interaction 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 Business 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 Siebel Servers can access it.

For further information on integration services and the Siebel MDA architecture, see "About Integration Services for Siebel Master Data Applications" on page 24 and other topics in Chapter 4, "Architecture and Framework."

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 5 lists the operation user properties for the UCM Transaction Manager.

Table 5. Operation User Properties for UCM Transaction Manager

User Properties	Value
IXMLOperation_ADD	EAI Siebel Adapter/Insert/
IXMLOperation_DELETE	EAI Siebel Adapter/Delete/

Table 5. Operation User Properties for UCM Transaction Manager

User Properties	Value
IXMLOperation_QUERY	EAI Siebel Adapter/Query/
IXMLOperation_QUERY_BY_ID	EAI Siebel Adapter/Query/PrimaryRowId;!SiebelMessage;
IXMLOperation_QUERYPAGE	EAI Siebel Adapter/QueryPage/ NewQuery=>IXML_NewQuery;SearchSpec=>SearchSpec;StartRowNum=>IXML_StartRowNum;PageSize=>IXML_PageSize;SERVICE_VALUE_RETURN;$LastPage=>IXML_LastPage;$NumOutputObjects=>IXML_NumOutputObjects;
IXMLOperation_SYNCH	EAI Siebel Adapter/Synchronize
IXMLOperation_UPDATE	EAI Siebel Adapter/Update
IXMLOperation_UPSERT	EAI Siebel Adapter/Upsert/
IONameForQueryBV_Account	CIFAccount
IONameForQueryBV_Contact	CIFContact
IONameForQueryBV_FINCORP Account	CIFFINCORPAccount
Account Address Field 1	Street Address;Street Address
Account Address Field 2	City;City
Account Address Field 3	State;State
Account Address Field 4	Postal Code;Postal Code
Account Address Field 5	Country;Country
Account Address Primary CUT Address	Primary Address Id;Account_Business Address
Contact Address Field 1	INS Personal Street Address;Street Address
Contact Address Field 2	INS Personal City;City
Contact Address Field 3	INS Personal State;State
Contact Address Field 4	INS Personal Postal Code;Postal Code
Contact Address Field 5	INS Personal Country;Country
Contact Address Primary Personal Address	Primary Personal Id;Contact_INS Personal Address
DispatcherMapName	CIFDispMap
Enable Updating SDH Type on Error	TRUE
EnableAgentLock	TRUE
Insert_IOandOp_Account	CIFAccountInsert;CIFAccountInsertRs;IXMLOperation_ADD

Table 5. Operation User Properties for UCM Transaction Manager

User Properties	Value
Insert_IOandOp_Contact	CIFContactInsert;CIFContactInsertRs;IXMLOperation_ADD
Insert_IOandOp_FINCORP Account	CIFFINCORPAccountInsert;CIFFINCORPAccountInsertRs;IXMLOperation_ADD
Operation Delete	IXMLOperation_DELETE
Operation Insert	IXMLOperation_ADD
Operation Update	IXMLOperation_UPDATE
Operation Upsert	IXMLOperation_UPSERT
Primary Object 1	Address/CIFPersonalAddress
SDHCommitSize	5
Update_IOandOp_Account	CIFAccountUpdate;CIFAccountUpdateRs;IXMLOperation_UPDATE
Update_IOandOp_Contact	CIFContactUpdate;CIFContactUpdateRs;IXMLOperation_UPDATE
Update_IOandOp_FINCORP Account	CIFFINCORPAccountUpdate;CIFFINCORPAccountUpdateRs;IXMLOperation_UPDATE
Upsert_IOandOp_Account	CIFAccountUpsert;CIFAccountUpsertRs;IXMLOperation_UPSERT
Upsert_IOandOp_Contact	CIFContactUpsert;CIFContactUpsertRs;IXMLOperation_UPSERT
Upsert_IOandOp_FINCORP Account	CIFFINCORPAccountUpsert;CIFFINCORPAccountUpsertRs;IXMLOperation_UPSERT

UCM Converter

Table 6 displays the only user property you can configure for this business service. This value appears in the preheader section of your CRMML message.

Table 6. 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 want the converter to error out.

UCM Dispatcher

You can modify both user properties for this business service as shown in Table 7.

Table 7. 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 8 displays these user properties.

Table 8. User Properties for UCM Security Manager

Name	Value
IXMLOperation_ADD	Insert
IXMLOperation_DELETE	Delete
IXMLOperation_QUERY	Query
IXMLOperation_QUERYPAGE	Query
IXMLOperation_UPDATE	Update
IXMLOperation_UPSERT	Upsert

UCM Publish/Subscribe Service

The user properties for this business service appear in Table 9.

Table 9. User Properties for UCM Publish/Subscribe Service

Name	Value
EnableCrossReference	TRUE or FALSE
WorkflowName	UCM Async Real-time Publish Workflow
ReloadSystemsCounter	This parameter determines when the business service reloads the system information (any changes) to its system cache (list) after that many iteration. Default value is 10. If set to 0, system information is never reloaded; if set to 1, system information is set after every iteration.

UCM Data Quality Manager

The user properties for this business service appear in Table 10.

Table 10. UCM Data Quality Manager User Properties

Name	Value
Account Account Cleansing Field 1	Name:Name;Location:Location;
Account Account Matching Field 1	Name:Name;Location:Location;
Account Account_Business Address Cleansing Field 1	Street Address:Street Address;City:City;State:State;Postal Code:Postal Code;Country:Country;
Account Account_Business Address Matching Field 1	Street Address:Street Address;City:City;State:State;Postal Code:Postal Code;Country:Country;
Account Auto Threshold	90
Account Cleansing Component 1	Account:Account;Account_Business Address:CUT Address;
Account Manual Threshold	70
Account Matching Component 1	Account:Account;Primary Account_Business Address;
Contact Auto Threshold	90
Contact Cleansing Component 1	Contact:Contact;Contact_INS Personal Address:Personal Address;Contact_Account:Account;
Contact Contact Cleansing Field 1	Last Name:Last Name;First Name:First Name;Middle Name:Middle Name;Job Title:Job Title;
Contact Contact Matching Field 1	Last Name:Last Name;First Name:First Name;Middle Name:Middle Name;
Contact Contact_Account Cleansing Field 1	Account:Account;Account Location:Location;
Contact Contact_Account Matching Field 1	Account:Account;Account Location:Location;
Contact Contact_INS Personal Address Cleansing Field 1	INS Personal Street Address:Street Address;INS Personal City:City;INS Personal State:State;INS Personal Postal Code:Postal Code;INS Personal Country:Country;
Contact Contact_INS Personal Address Matching Field 1	INS Personal Street Address:Street Address;INS Personal City:City;INS Personal State:State;INS Personal Postal Code:Postal Code;INS Personal Country:Country;
Contact Manual Threshold	70

Table 10. UCM Data Quality Manager User Properties

Name	Value
Contact Matching Component 1	Contact:Contact;Primary Contact_INS Personal Address;Contact_Account;
ExactMatch Object 1	Account:CIFAccount;Contact:CIFContact;

UCM Batch Manager

The user properties for this business service appear in Table 11.

Table 11. User Properties for UCM Batch Manager

Name	Value
Account AutoMatch	CIFAccountUpdate;CIFAccountUpdateRs;IXMLOperation_UPDATE
Account NoMatch	CIFAccountInsert;CIFAccountInsertRs;IXMLOperation_ADD
Contact AutoMatch	CIFContactUpdate;CIFContactUpdateRs;IXMLOperation_UPDATE
Contact NoMatch	CIFContactInsert;CIFContactInsertRs;IXMLOperation_ADD
FINCORP Account NoMatch	CIFFINCORPAccountInsert;CIFFINCORPAccountInsertRs;IXMLOperation_ADD

UCM Survivorship Engine

The user properties for this business service appear in Table 12.

Table 12. User Properties for UCM Survivorship Engine

Name	Value
DefaultAttrGrpName_Account	Default Account Attribute Group
DefaultAttrGrpName_Contact	Default Contact Attribute Group
ExemptFields_Account	Party UId;Id
ExemptFields_Contact	Party UId;Person UId;Id

UCM Account Source Data and History Service

The user properties for this business service appear in Table 13.

Table 13. User Properties for UCM Account Source Data and History Service

Name	Value
Application Services Interface	Y
Internal Integration Object	UCMAccountSourceDataAndHistory

UCM Contact Source Data and History Service

The user properties for this business service appear in Table 14.

Table 14. User Properties for UCM Contact Data and History Service

Name	Value
Application Services Interface	Y
Internal Integration Object	UCMContactSourceDataAndHistory

Configuring UCM Transaction Manager for Data Management and Survivorship

Enable customer data management or the survivorship engine for UCM Transaction Manager business service by configuring workflow input arguments.

For details on configuring input arguments using Siebel Tools, see *Siebel Business Process Designer Administration Guide*.

For background information on UCM Transaction Manager business service, see "UCM Transaction Manager" on page 48.

To configure UCM Transaction Manager to Enable Data Management and Survivorship Engine

- Configure the following input arguments for the UCM Transaction Manager:

Input Arguments	Default Value
TurnOnCDMCleanse	FALSE
TurnOnCDMExactMatch	FALSE
TurnOnCDMMatch	FALSE
TurnOnSE	FALSE

Configuring UCM Auto Merge and Manual Review Threshold Values

Configure Universal Customer Master (UCM) auto merge and manual review threshold functionality by configuring user properties for the UCM Data Quality Manager business service.

The auto threshold value determines which incoming records are merged into existing matching best version records. The manual threshold determines the value (between auto merge and manual review value) at which incoming records are shown to administrators in Data Management screens for review. For information on manually reviewing records in the data management screen, see "Managing Pending Updates from External Systems" on page 93.

For details on configuring business service user properties using Siebel Tools, see *Integration Platform Technologies: Siebel Enterprise Application Integration*.

For background information on UCM Data Quality Manager business service, see "UCM Publish/ Subscribe Service" on page 51.

To configure auto merge and manual review thresholds

- Configure the following user properties for the UCM Data Quality Manager business service:

User Property	Value
Account Auto Threshold	90
Account Manual Threshold	70
Contact Auto Threshold	90
Contact Manual Threshold	70

Configuring UCM Survivorship Engine Properties

Configure the following Universal Customer Master (UCM) survivorship functionality by configuring user properties in UCM Survivorship Engine business service:

- Default attribute group names
- Fields exempt by survivorship rules engine

For details on configuring business service user properties using Siebel Tools, see *Integration Platform Technologies: Siebel Enterprise Application Integration*.

For background information on UCM Survivorship Engine business service, see "UCM Survivorship Engine" on page 53.

To configure survivorship properties

- Configure the following user properties for the UCM Survivorship Engine business service:

User Property	Default Value
DefaultAttrGrpName_Account	Default Account Attribute Group
DefaultAttrGrpName_Contact	Default Contact Attribute Group
ExemptFields_Account	Party UId; Id
ExemptFields_Contact	Party UId; Person UId; Id

About UCM Data Management Services

Siebel Universal Customer Master (Siebel UCM) provides the following data management services to cleanse, identify, and link incoming source data to the master data records. These services are invoked through the Siebel Enterprise Application Integration (Siebel EAI) interface and can process single or multiple records. Records updated directly within Siebel UCM application also trigger similar data management services but do not trigger survivorship rules.

- Account Data Management Service
- Contact Data Management Service

These services take an incoming Account or Contact record and:

- Check the system privilege of the application submitting the record
- Cross-reference the record if external account ID is provided
- Create the UUID for the record if the record is new
- Call the data cleansing engine to standardize and validate account or contact name and address
- Perform an exact match process based on configured parameter (default is external account ID for account records; external contact ID for contact records).
- If no exact match found, call data matching engine to perform fuzzy matching (default is Account name for account records; default is Contact first name and last name for contact records) to further identify possible duplication.

NOTE: Siebel UCM provides embedded data matching and cleansing third-party software (SSA). The Siebel Data Quality module is licensed separately from Siebel UCM. See *Siebel Data Quality Administration Guide* and your third-party data quality provider documentation for further information. See also, "Configuring Siebel Data Quality Cleansing for Siebel UCM" on page 97.

Depending on the matching score returned from the data matching engine, UCM can create a new record, link to an existing record, or store the incoming record for further investigation by the data steward. By default the data matching engine returns two threshold scores, an auto threshold score (upper) and manual threshold scores (lower), numbered on a scale of 0-100. These threshold numbers are configurable using Siebel Tools. The matching score results in the following three scenarios:

- When the incoming record is matched above the upper threshold, Siebel UCM applies survivorship rules to merge the incoming record with an existing record in Siebel UCM. For information on survivorship and details on setting up the rules, see "Process for Creating Survivorship Rules" on page 83. When the incoming record is matched below the lower threshold, Siebel UCM creates a new record and publishes a new-record message to other external systems. For information on publishing and subscribing, see "Configuring System Publish and Subscribe for Siebel Master Data Applications" on page 79.
- When the incoming record is matched in between the upper and the lower threshold numbers, Siebel UCM stores the record in the deduplication table (S_UCM_DEDUP) for further investigation. Data stewards can view the pending records in the Suspect Cases-Account or Contact screen.
- Configure your matching threshold and the data engine matching rules based on your organization's data quality standards.

Configuring UCM Data Quality Manager for Exact Match

Configure the Universal Customer Master (UCM) exact match functionality by configuring user properties in the UCM Data Quality Manager business service.

For details on configuring business service user properties using Siebel Tools, see *Integration Platform Technologies: Siebel Enterprise Application Integration*.

Siebel UCM Exact Match functionality has the following characteristics:

- If user property Enable XRef Match is set to TRUE, Exact Match looks up the cross reference database to find a match, otherwise, it skips this step and continues with the next step (either QueryByExample or QueryByUserKey).
- If a Name ExactMatch Field user property is specified, Exact Match will QueryByUserKey, using the Integration Object specified in ExactMatch Object User Property.
- Exact Match performs either QueryByExample or QueryByUserKey, but not both.
- To improve throughput during initial loading of data, se the Enable XRef Match to FALSE.
- If you configure exact match fields to overwrite the default preconfigured behavior, make sure the fields can be used to uniquely identify a record.

To configure Siebel UCM exact match

- Configure the following user properties for the UCM Data Quality Manager business service:

User Property	Value
ExactMatch Object	{Name + ":" + IntObjName + ";"}
Name ExactMatch Field	{ICFieldName + ":" + BCFieldName + ";"}
Enable XRef Match	TRUE or FALSE

where:

Name—Should be unique; it could be an integration object name, root component name, or supported objects (for example, Account, Contact).

IntObjName—Integration object name (use for QueryByUserKey).

ICFieldName—Integration component field name.

BCFieldName—Business component field name (use for QueryByExample).

Configuring UCM Unmerge Functionality

Configure child objects for Universal Customer Master (UCM) unmerge functionality by configuring user properties for the UCM DeDuplication Results (Account) or UCM DeDuplication Results (Contact) business components. This configuration determines the list of child objects that are reparented to the appropriate record during an unmerge execution.

For background information on configuring business components in Siebel Tools, see *Configuring Siebel Business Applications*.

To configure child objects for UCM unmerge functionality

- Configure the default values for the following business component user properties:

Business Component	User Property	Default Value
UCM DeDuplication Results (Account)	MERGE_SUPPORT_CHILD0	CUT Address;FINCORP Loan Account;FINCORP Deposit Account;FIN Business Phone;UCM Account Source Data and History;CIF Account Reference
UCM DeDuplication Results (Contact)	MERGE_SUPPORT_CHILD0	Personal Address;FINCORP Account;FIN Contact Phone;UCM Contact Source Data and History;CIF Contact Reference

Configuring UCM Server Components for Batch Transactions

Configure the UCM server components UCM Batch Manager and UCM Batch Publish Subscribe for batch transactions by setting parameter values for these components. Parameter values can be set in the Server Manager GUI or at the command-line interface. For full details on setting server component parameters, see *Siebel System Administration Guide*.

Batch Processing performs matching of the incoming record against the existing records in Siebel UCM. Adjust the Commit Size and Batch Size parameters to higher values for optimal batch operation.

If there are possible duplicates within the same batch or commit size, UCM Batch Process does not identify them automatically. These exceptional duplicates are captured when the Siebel Data Quality De-duplication server component is run. After running this component, the exceptional duplicates are captured and displayed within UCM Conflict Resolution view, where data stewards can review and merge the records as appropriate. Any updates through the UI that result in duplications can also be managed through the same Data Quality De-duplication process.

For information on running batch transactions, see "Running Data Management in Batch Mode" on page 95.

For background information on UCM server components, see "About Siebel Universal Customer Master Server Components" on page 37.

To configure UCM Server Components for Batch Transactions

- Configure the following parameters for the appropriate UCM server component:

Parameter	Alias	Default Value	Description
UCM Survivorship Engine Flag	UCMSurvivorshipEngineFlag	FALSE	To enable the UCM survivorship engine capability
UCM Data Management Flag	UCMDataMgmtFlag	FALSE	To enable the UCM data management capability
UCM Publish/ Subscribe Flag	UCMPubSubFlag	FALSE	To enable the UCM publish and subscribe capability
UCM Batch Object Type	UCMBatchObjectType	Contact	Object type for UCM Batch Publish Subscribe or UCM Batch Manager server components
UCM Batch Size	UCMBatchSize	10	Number of records to be processed at one time
UCM Sleep Time	UCMSleepTime	60	Sleep Time (seconds)

Configuring Run-time Events for UI Interaction

Creating UCM run-time events in the Administration - Runtime Events screen is necessary to configure enhanced UI interactions. For background information on run-time events and for details on creating these events, see *Siebel Personalization Administration Guide*.

To configure run-time events for UI interaction

1 Navigate to Administration - Runtime Events > Action Sets view, and create the following action sets:

Action Set Name	Activate	Enable Export
UCM WriteRecord	Y	Y
UCM PreWriteRecord	Y	Y
UCM PreDeleteRecord	Y	Y
UCM RecordDeleted	Y	Y

2 For each action set defined, create one new record in the Action Set's more info view with the following information that corresponds with the action set name (for example, the WriteRecord business service method is the record for the UCM WriteRecord action set):

Name	Business Service Name	Business Service Method	Action Type	Sequence
UCM WriteRecord	UCM UI Interaction Service	WriteRecord	BusService	1
UCM PreWriteRecord	UCM UI Interaction Service	PreWriteRecord	BusService	1
UCM PreDeleteRecord	UCM UI Interaction Service	PreDeleteRecord	BusService	1
UCM RecordDeleted	UCM UI Interaction Service	RecordDeleted	BusService	1

3 Navigate to Administration - Runtime Events > Events, and create four events with the following field information:

Object Type	Object Name	Event	Subevent	Action Set Name	Seq-uence
Applet	Contact List Applet	InvokeMethod	WriteRecord	UCM WriteRecord	1
Applet	Contact List Applet	PreInvokeMethod	WriteRecord	UCM PreWriteRecord	1

Object Type	Object Name	Event	Subevent	Action Set Name	Seq-uence
Applet	Contact List Applet	InvokeMethod	DeleteRecord	UCM RecordDeleted	1
Applet	Contact List Applet	PreInvokeMethod	DeleteRecord	UCM PreDeleteRecord	1

4 Create the same four events for the following additional screen views (objects):

- SIS Account Entry Applet
- SIS Account List Applet
- Contact Form Applet

Configuring Default System for UI Interaction

A default system must be registered in the System Registration view to enable UI interaction.The Default System is necessary to capture the registered system that last modified the data for future Survivorship Engine evaluation. Depending on UCM deployment the default system can be UCM or UCM and a CRM combined instance.

To configure a default system for UI interaction

- Register the system in the System Registration view with the following data:
  - System ID = Default System
  - System Name = Default System

  For further information on this task, see "Registering Systems Connected to Siebel Master Data Applications" on page 73.

Configuring UCM Inbound Server Communication

Configuring inbound UCM Siebel Server communications depends on the registered system protocol type for the Siebel Connector for Master Data Applications, either the Siebel EAI MQSeries Transport or the Siebel EAI HTTP Transport. (For further details on the Siebel Connector for Master Data Applications, see "Siebel Connector and Integration of Siebel Master Data Applications" on page 30.)

The Siebel EAI MQSeries Transport allows you to integrate data between Siebel Business Applications and external applications that can interface with the IBM MQSeries. The EAI MQSeries Server Transport transports messages to and from IBM MQSeries queues. To configure Siebel EAI MQSeries Transport, see "Configuring for Siebel EAI MQSeries Transport" on page 62.

Siebel EAI HTTP Transport allows you to send XML messages over HTTP to a target URL. The Siebel Web Engine (SWE) serves as the transport to receive XML messages sent over the HTTP protocol to a Siebel application. To configure Siebel EAI HTTP Transport, see "Configuring Siebel EAI HTTP Transport" on page 64.

For background information and details on configuring Siebel EAI MQSeries Transport and Siebel EAI HTTP Transport, see *Transports and Interfaces: Siebel Enterprise Application Integration*.

Configuring for Siebel EAI MQSeries Transport

To configure UCM inbound communication using the Siebel EAI MQSeries Transport, you must:

- Create and configure two named subsystems.
- Configure the Siebel Server component MQSeries Server Receiver (alias: MqSeriesSrvRcvr)

NOTE: Make sure Siebel EAI MQSeries Transport is enabled prior to configuring for UCM. For further information on enabling Siebel EAI MQSeries Transport, see *Transports and Interfaces: Siebel Enterprise Application Integration*.

For background information on these Siebel Server administrative tasks, see *Siebel System Administration Guide*.

To configure named subsystems

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 check mark 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*.

To configure MQSeries Server Receiver Siebel Server component

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 subsystems" on page 62. 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.

Configuring Siebel EAI HTTP Transport

To configure UCM inbound communication using the Siebel EAI HTTP Transport, you must:

- Configure the Siebel Web Engine (SWE)
- Modify the Siebel Universal Customer Master Configuration file (`.cfg` file) to run the extension service.

NOTE: Make sure Siebel EAI HTTP Transport is enabled prior to configuring for UCM. For further information on enabling Siebel EAI HTTP Transport, see *Transports and Interfaces: Siebel Enterprise Application Integration*.

For background information on these Siebel Server administrative tasks, see Siebel System Administration Guide.

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:

   ```
   [/ucm_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 more information on this procedure, see *Transports and Interfaces: Siebel Enterprise Application Integration*.

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

1. Open `ucm.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.

Configuring Siebel UCM List of Values

Configure Siebel Universal Customer Master (Siebel UCM) list of values in the Administration - Data > List of Values screen. For background information on list of values and for detailed instructions, see *Applications Administration Guide*.

To configure Siebel UCM List of Values

1. Navigate to Administration - Data > List of Values.
2. Create the following new List of Values records.

Type	Display Value	Language-Independent Code	Order
UCM_SDH_TYPE_MLOV	Source-Insert	Source-Insert	12
UCM_SDH_TYPE_MLOV	Source-Update	Source-Update	14
UCM_SDH_TYPE_MLOV	Source-Upsert	Source-Upsert	16
UCM_SDH_TYPE_MLOV	Queued-Insert	Queued-Insert	62
UCM_SDH_TYPE_MLOV	Queued-Update	Queued-Update	64
UCM_SDH_TYPE_MLOV	Queued-Upsert	Queued-Upsert	66
CIF_PROTOCOL_XCHG_TYPE	JMS	JMS	15

3. Make sure the following settings are additionally set for each record:
   - Language Name = English-American
   - Active = TRUE (checked)
   - Translate = TRUE (checked)
   - Replication Level = All

Configuring Deep Copy for Data Hierarchies in UCM

A deep copy of a data hierarchy copies not only the hierarchy header but all the relationships in the hierarchy as well. To configure this functionality set the following user properties for the UCM Hierarchy business component.

For background information on configuring business components, see *Configuring Siebel Business Applications*.

To configure deep copy for data hierarchies

- Set the following user properties for the UCM Hierarchy business component:

Name	Value	Description
Deep Copy	UCM Hierarchy Relationship	Add this property to configure a deep copy of the UCM Hierarchy Relationship.
Deep Copy 1	UCM Hierarchy Reporting Relationship	(Optional) If specified, the denormalized relationship stored in UCM Hierarchy Reporting Relationship will be copied as well. If not specified, a new denormalized relationship will be generated based on the relationship in UCM Hierarchy Relationship.
DoValidation	Y	(Optional) If specified, will invoke ValidateHierarchy before deep copy to check the validity of the hierarchy relationship and to make sure there are no broken parent chains in the relationship. If a broken parent chain is identified, an error message occurs and deep copy exits.

About Siebel UCM and CRM Integration

Siebel Universal Customer Master (Siebel UCM) is preconfigured to support integration with a Siebel CRM Business Application and Siebel UCM. When enabled, the integration allows Siebel UCM to apply history, cross-reference, and data quality processes to Siebel CRM contact and account data.

The integration of Siebel UCM and CRM uses the Siebel UCM CRMML interface to deliver and receive data messages, and includes the following preconfigured workflows to facilitate the integration:

- CRM-UCM-Account-WF
- CRM-UCM-Contact-WF
- UCM-CRM-Account-WF
- UCM-CRM-Contact-WF

For information on extending the Siebel UCM and CRM integration, see "Extending Siebel UCM and CRM Integration" on page 66.

To enable the preconfigured Siebel UCM and CRM integration, follow the instructions in `crm782-ucm782-intgn-readme.txt`, and make sure the system preference Enable UCM Processes is set to TRUE. For further information on UCM System Preferences, see "About System Preferences for Siebel UCM" on page 96.

Extending Siebel UCM and CRM Integration

Extending the preconfigured Siebel UCM and CRM integration to accommodate corresponding UCM or CRM system configurations and extensions requires the following high-level configurations:

- Update the appropriate Integration Object for Siebel UCM, CRM, or both
- Generate a new XDS for the message body payload
- Update the Transformation XSLT
- Update the Validation XSLT if additional validation is needed

It is recommended to use a third-party standards-based XSLT editor. For further information on these tasks, see *XML Reference: Siebel Enterprise Application Integration*.

Configuring Siebel Master Data Integration Objects

This topic contains a list of general integration configurations you can perform after installing and initially configuring Siebel Master Data Applications.

See the following related topics for further information on other areas of application configuration and the installation of Siebel MDA.

Related Topics

"Installing Siebel Master Data Applications" on page 39

"Configuring Initial Setup for Siebel Master Data Applications" on page 45

"Configuring Siebel Master Data Application Features" on page 46

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"

For background information on integration objects, see *Integration Platform Technologies: Siebel Enterprise Application Integration*.

UCM Integration Objects

The UCM integration objects contain the data hierarchy that is a subset of the data in a Siebel application business object. The UCM Integration objects map to the different UCM business objects in the Siebel Business Application. Many preconfigured integration objects are provided for you, which you can extend and modify as necessary.

These integration objects can be found in the Siebel repository, and their names start with CIF. In addition, you can create your own integration objects for use using Integration Object Builder in Siebel Tools; see *Integration Platform Technologies: Siebel Enterprise Application Integration* for further information.

This internal integration object is required so that the UCM business services, such as the UCM Transaction Manager, can receive and package the data from XML (as a CRMML message) and into a format understood by the Siebel application object manager (as business component data).

UCM Envelope Integration Objects

The envelope integration object stores system information about where the data originated and where the destination of the data. It stores information about the message (such as its ID) rather than the message contents.

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. Otherwise, use the envelope integration object named UCM Envelope.

For information on customizing the envelope integration object, see "Modifying UCM Envelope Integration Objects" on page 69.

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.

Modifying Siebel Master Data Application Integration Objects

This topic describes the task of modifying Siebel Master Data Application integration objects.

For background information on Siebel Master Data Application integration objects, see "About Siebel Master Data Application Integration Objects" on page 67.

To modify UCM Integration Objects

1. Determine whether or not you need to customize any integration objects listed in Table 21 on page 111, 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 preconfigured integration objects, see Chapter 8, "Siebel Master Data Applications Messages."

Modifying UCM Envelope Integration Objects

This topic describes the task of modifying Siebel UCM envelope integration objects.

For background information on Siebel Master Data Application integration objects, see "About Siebel Master Data Application Integration Objects" on page 67.

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 Message" on page 30 and "About CRMML Messages" on page 116.

Extending UCM Integration Objects

This topic describes the task of extending the integration object to include other data, such as Assets, Service Requests, and so on. Extending the UCM integration objects to include other data requires the creation of a new integration object based on either the Account, Contact, or Household business object. Only objects that are based on the Party data model are supported by UCM specific services.

For background information on Siebel Master Data Application integration objects, see "About Siebel Master Data Application Integration Objects" on page 67.

To modify UCM Integration Objects to include other data

1. From Siebel Tools, choose Object Explorer > Integration Object.
2. Create an integration object with integration components based on either the Account, Contact, or Household business object. For example, create an integration object with integration components Account and Assets based on the Account business object.

For background information on integration objects and how to create them, see *Integration Platform Technologies: Siebel Enterprise Application Integration*.

Modifying the Map Property of the Master Data Application Integration Objects

This topic describes the task of modifying the map property of Siebel Master Data Application dispatcher map integration objects.

For background information on Siebel Master Data Application integration objects, see "About Siebel Master Data Application Integration Objects" on page 67.

To modify 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; modify the properties as appropriate.

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

```
CRMML/PartyPackage/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/PartyPackage/
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 Table 15:

Table 15. Dispatcher Map Value User Property

Token	Description	Example
1st	The location to insert the remaining five tokens at runtime **NOTE:** The message needs to start with CRMML/PartyPackage because the UCM Dispatcher uses it to inspect the incoming XML data; that is, the first token is an XPath (XML Path Language) search expression.	`CRMML/PartyPackage/DeleteAccount`
2nd	System reserved token	a
3rd	System reserved token	b
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.

Adding Status Key Property to UCM Integration Object

This topic describes the task of adding the Status Key property to Universal Customer Master (UCM) integration objects.

For background information on Siebel Master Data Application integration objects, see “About Siebel Master Data Application Integration Objects” on page 67.

To add status key property to UCM integration object

1 Make sure that there is no active StatusObject user property in the UCM Transaction manager business service.

For background information on UCM Transaction Manager, see “UCM Transaction Manager” on page 31.

2 Modify the input integration object to reflect the status key you require at each integration component level. For further information on this task, see *Integration Platform Technologies: Siebel Enterprise Application Integration*.

3 Set the Siebel Adapter user property StatusObject = True using Siebel Tools, and compile the Siebel repository file (srf).

7 Administering Siebel Master Data Applications

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

This chapter includes the following topics:

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. After registering, the system's privileges and accessibility to the Siebel Master Data Applications is administered on other tabs within this view. For information on further defining the registered system, see:

- "Setting System Privileges for Siebel Master Data Applications" on page 76
- "Configuring System Publish and Subscribe for Siebel Master Data Applications" on page 79

To create a new system

1. From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.
2. From the link bar, select System Registrations.
3. In the System Registrations view, click New.
4. Enter the appropriate information in the available fields to define the system.

   Use Table 16 for information on each of the fields.

Table 16. System Fields

Field	Description
System ID	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, MQSeries, or JMS.
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.
URL	URL destination for posting responses to messages. Only applicable when protocol type is HTTP.
Connection Factory	The name of the connection factory; that is, the connection means to the JMS provider. Only applicable when protocol type selected is JMS.
Send Queue	Specifies the JMS queue that messages get sent to. Only applicable when protocol type selected is JMS.
Description	Description of the application accessing Siebel Master Data Applications.
Comment	Comments regarding the application accessing Siebel Master Data Applications.

Figure 7 shows the creation of an example system.

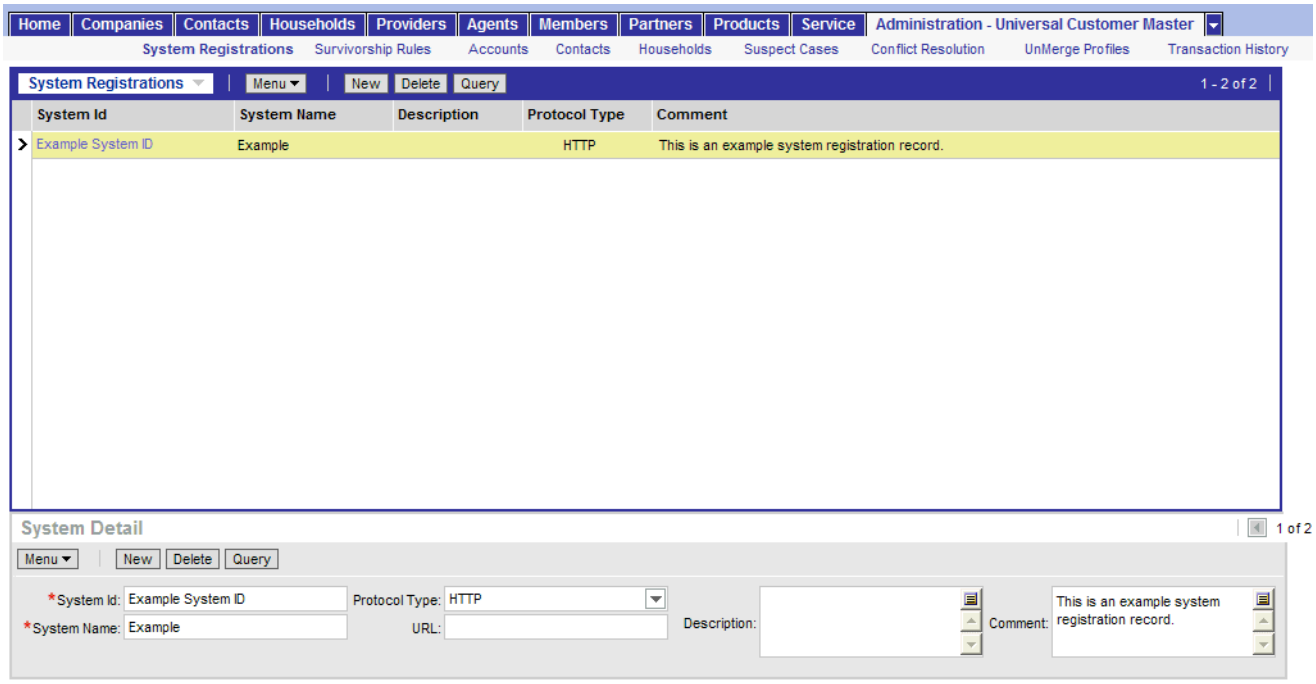

Figure 7. Creating a New System

To delete a system

1. From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.
2. From the link bar, select System Registrations.
3. 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.
4. In the System Registrations form, click Delete.

   A confirmation dialog appears.

   NOTE: Deletion of registered systems can create foreign key integrity violations due to referencing of external systems in cross reference, survivorship definition, source data, history records, and so on.
5. Select OK to delete the system.

Setting System Privileges for Siebel Master Data Applications

After a system is registered, 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.

For information on registering a system, see "Registering Systems Connected to Siebel Master Data Applications" on page 73.

To set a system's privileges

1. From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.
2. From the link bar, select System Registrations.
3. In the Systems Registrations list, select the system of interest, and drill down on the hyper link in the System ID field.
4. From the System Detail view, select the System Privileges tab.
5. In the System Privileges view, click New.
6. Enter the appropriate information in the available fields to define the privileges for each object in the system.

   Use Table 17 for information on each of the fields.

Table 17. 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.
Comment	Comments on the privileges chosen for the selected object of the selected system.

Figure 8 shows the definition of privileges for the example system.

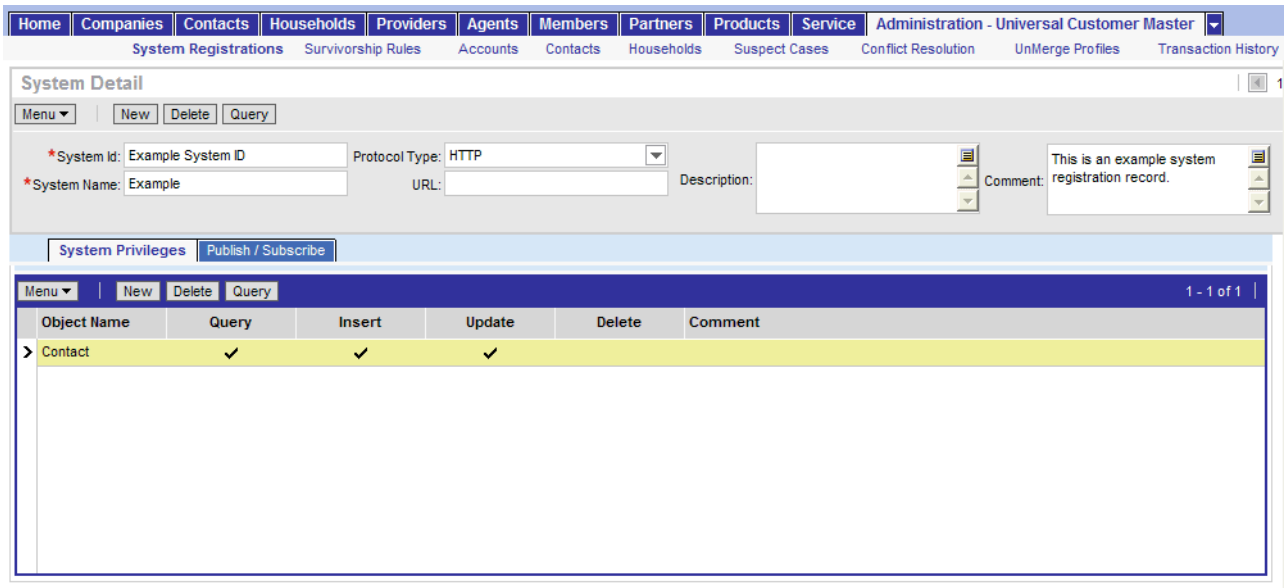

Figure 8. Defining System Privileges

The configuration in Figure 8 indicates that the system registered with the name Example has query, insert, and update privileges for the Contact business object. However, it is not allowed to delete any of the Contact records in the Siebel Universal Customer Master application. Furthermore, it is set to subscribe to record changes for the Contact business object.

About System Publish and Subscribe Modes

Siebel Master Data Applications uses one of three modes to publish or update data: real-time mode, batch mode, and event mode. Details on how these modes function within the application are as follows.

For details on configuring these modes, see "Configuring System Publish and Subscribe for Siebel Master Data Applications" on page 79.

Real-Time Mode

When configured, the real-time mode of the publish and subscribe functionality runs when another process or business service explicitly invokes the UCM Publish/Subscribe Service business service. The calling process sends this business service a record to publish. The following workflow processes provide examples of real-time publishing: The UCM Server Party Package, UCM Account Batch Data Management Workflow, and UCM Contact Batch Data Management Workflow.

Batch Mode

When configured, the batch mode of the publish and subscribe functionality runs similar to a background server component rather than a more traditional batch component. The UCM Batch Publish Subscribe (alias UCMBatchPubSub) component must be online while changes (inserts or updates, but not deletes) are being made to the target business component (for example, Contact, Account, and so on).

During a batch mode operation, the UCMBatchPubSub server component calls the UCM Daily Publish workflow process. This workflow process calls the UCM Publish/Subscribe Service business service, and this business service checks if it is being run on a server component. If the service is being run on a server component, it retrieves the server parameters for UCM Batch Object Type (alias UCMBatchObjectType), UCM Batch Size (alias UCMBatchSize), and UCM Sleep Time (alias UCMSleepTime). Then the business service iteratively runs through the following steps until the server component stops:

1. Creates an internal search specification, SearchSpec, for the business component based on the following format:

   `'(([Last Update - SDQ] <= 'mm/dd/yyyy hh:mm:ss') AND ([Last Update - SDQ] > 'mm/dd/yyyy hh:mm:ss'))'`

   where:

   `mm/dd/yyyy hh:mm:ss` arguments = the current time and the current time less the UCMSleepTime parameter value (default value of 60 seconds).

   `[Last Update - SDQ]` = the business component field name of the integration component field name specified by the SearchFieldName user property configured in the UCMBatchObjectType integration object. By default, the IO CIFContact has the IC property SearchFieldName = Last Update, which maps to the BC Contact's field Last Update - SDQ component field value [Last Update - SDQ]

   NOTE: You cannot create a custom search specification.

2 The search specification is passed to the EAI Siebel Adapter, and it uses the method QueryPage with the SearchSpec against the UCMBatchObjectType to find the records that have changed or are new. The UCMBatchSize parameter value is used to determine how many records are returned in the query page's result set.

3 For each record returned in the query page's result set, it is published according to the registered UCM systems that have the object defined in the Publish/Subscribe view, that have a publish frequency set to Batch, and that are not expired.

Event Mode

Any system configured for Real-time or Batch mode can also subscribe to Event mode if that system requires the message to be published when an event occurs. Events refer to Siebel UCM operations such as Merge or Unmerge or Link and Update or Create New and so on. Though preconfigured, these events do not get published; Siebel UCM can be customized to prepare the outbound message (resulting from such operations) by using UCM Publish/Subscribe Service - EventPublishMethod.

Configuring System Publish and Subscribe for Siebel Master Data Applications

The Publish and Subscribe infrastructure provides workflow policies, workflow processes, 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. The messages are predefined and used for the publication of records flagged by the workflow policies.

The following procedures describe:

- Configuring real-time publish and subscribe for a system
- Configuring batch-time publish and subscribe for a system

To configure real-time publish and subscribe for a system

1 From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.

2 From the link bar, select System Registrations.

3 In the System Registrations list, select the system of interest, and drill down on the hyper link in the System ID field.

4 From the System Detail view, select the Publish/Subscribe tab.

5 In the Publish/Subscribe view, select an existing record to configure or click New.

6 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. For the batch-time procedure, see "To configure batch-time publish and subscribe for a system" on page 81.
Last Published	The last time the Siebel Master Data Applications published record changes for the selected object to this selected system.
Event	Signifies if the object subscribes to event publishing.
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.

7 From the application-level menu, choose Navigate > Site Map > Administration - Business Process screen.

8 From the link bar, select Workflow Processes.

9 Query for UCM Server Party Package Workflow or a prepared customized workflow.

For further information on the UCM Server Party Package Workflow, see "About Siebel Master Data Application Workflows" on page 34.

10 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 and configuring workflow processes, see *Siebel Business Process Designer Administration Guide.*

Figure 9 displays the definition of real-time publish and subscribe capabilities for the example system.

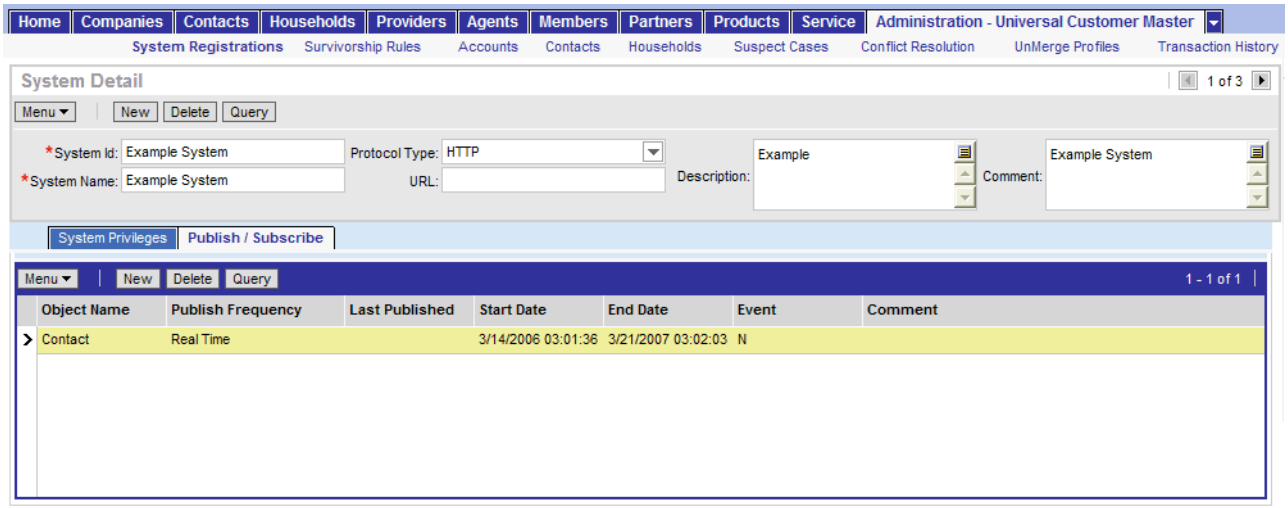

Figure 9. Defining System Real-time Publish and Subscribe Capabilities

To configure batch-time publish and subscribe for a system

1. From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.
2. From the link bar, select System Registrations.
3. In the System Registrations list, select the system of interest, and drill down on the hyper link in the System ID field.
4. From the System Detail view, select the Publish/Subscribe tab.
5. In the Publish/Subscribe view, select an existing record to configure or click New.
6. 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. For the real-time publish and subscribe procedure, see "To configure real-time publish and subscribe for a system" on page 79.

Field	Description
Publish Time	Selects at which time the update occurs. Only applicable when Publish frequency is Daily Batch.
Event	Signifies if the object subscribes to event publishing.
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.

7 Step off the record to save the system publish and subscribe functionality.

Figure 10 displays the definition of batch-time publish and subscribe capabilities for the example system.

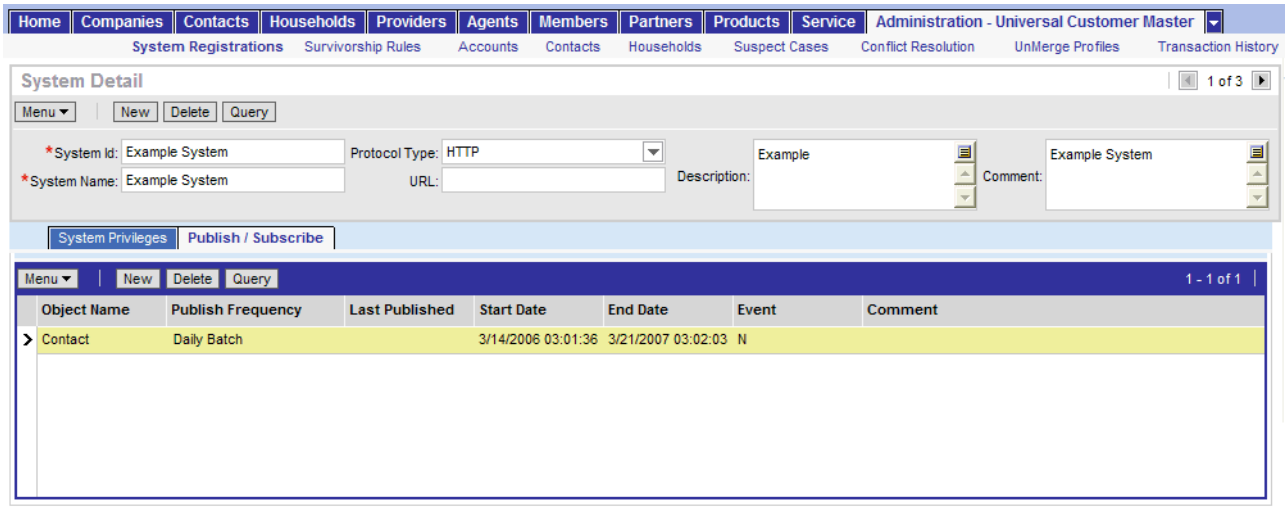

Figure 10. Defining System Batch-time Publish and Subscribe Capabilities

About UCM Survivorship Rules

Survivorship rules are an automated means of controlling the quality of customer data stored in the Universal Customer Master (UCM). Multiple systems connect to the UCM with record update and insert requests, and, to make sure that UCM presents the most trusted information contributed by each of these systems, administrators can create and use survivorship rules to govern which updates from which external systems can be trusted at the field level based on key criteria.

Survivorship rules use attribute groups, which determine the set of fields to be evaluated by the survivorship rule, and use comparison criteria, such as confidence level of the publishing system or the most recent published data, to evaluate whether inputs from given external systems can be used to update the UCM.

There is an implicit default attribute group, which includes the fields that are not explicitly defined in an attribute group definition. There is also a default criteria for each survivorship rule, that is, if there is no explicit rule definition for a certain attribute group (including the default attribute group), then the default criteria applies to determine whether an external system can update the UCM.

NOTE: Do not modify or delete the names of the default attribute groups for Account and Contact. If these names are modified, make sure they match the values in the UCM Survivorship Engine User Properties: DefaultAttrGrpName_Account or DefaultAttrGrpName_Contact.

There are three comparison methods used by survivorship rules to evaluate the quality of the object fields associated with an attribute group:

- Recent - Compares the date of the object's attribute group field's data against the date of the updating message. The most recent data survives in the record.
- History - Compares the date of the object's attribute group field's data against the date of the updating message. The oldest data survives in the record.
- Source - Compares the confidence level of the external system that contributed data to attribute group fields in the object against the confidence level of the external system of the updating message. The data of the highest confidence level survives in the record.

Note the following restrictions and recommendations when defining survivorship rules:

- Although administrators can define multiple rules for each object (account, contact), use only one active rule per object.
- Each individual field within a given object can only be included in one attribute group definition, otherwise conflicts may occur when survivorship rules are evaluated.

For details on the process of creating survivorship rules, see "Process for Creating Survivorship Rules" on page 83.

Process for Creating Survivorship Rules

This topic lists the ordered tasks of the process for creating survivorship rules for Master Data Applications. For background information on survivorship rules, see "About UCM Survivorship Rules" on page 82.

Perform the following tasks to create survivorship rules:

1. Create Attribute Groups for the Survivorship Rule. For information on this task, see "Creating Attribute Groups for Survivorship Rules" on page 84.
2. Create the Survivorship Rule. For information on this task, see"Creating UCM Survivorship Rules" on page 84.
3. Activate and test the Survivorship Rule. For information on this task, see "Activating UCM Survivorship Rules" on page 86.

Creating Attribute Groups for Survivorship Rules

Creating attribute groups for survivorship rules is a task in the "Process for Creating Survivorship Rules" on page 83. For further information on survivorship rules, see "About UCM Survivorship Rules" on page 82.

To create an attribute group

1. From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.
2. From the link bar, select Survivorship Rules.
3. Select the Attribute Group Fields tab.
4. In the Attribute Group list, click the New button.
5. Enter a descriptive name for the attribute group record and select an object name.
6. In the Attribute Group Field Name list, click New.
7. Add one or more fields to define this attribute group.

Creating UCM Survivorship Rules

Creating survivorship rule records is a task in the "Process for Creating Survivorship Rules" on page 83. For further information on survivorship rules, see "About UCM Survivorship Rules" on page 82.

To create a survivorship rule

1. From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.
2. From the link bar, select Survivorship Rules.
3. In the Rule Name list, click New.

4 Enter values for the columns in the record. See the following table for details on the available columns.

Column Name	Description
Object Name	Use this column to designate what objects are affected by the survivorship rule. For example, select the contact object to create a survivorship rule based on contact information.
Default Criteria	Use this column to configure the default criteria used for attribute groups with no explicit rule definition. It is also the default comparison rule to be used when the comparison rule at the Attribute Group level cannot be used to determine the winning data. For example, when comparing by source and the confidence level of the systems are tied, the survivorship rule uses the Default Criteria as the comparison rule to compare.
Status	Sets the survivorship rule to active or inactive. Select inactive while creating and defining the survivorship rule.
Default Rule Set	When no active survivorship rule is defined or the active survivorship rule has expired, the rule with the Default Rule Set set to Y will be used as the active survivorship rule. Only check Default Rule Set for one survivorship rule per object.
Start Date	Configures the start date at which the survivorship rule becomes valid.
End Date	Configures the end date at which the survivorship rule is invalidated.

5 In the Attributes Group list click the new button to add attribute groups explicitly governed by the survivorship rule.

6 From the Attributes Group pick list, select the attribute group of focus for this survivorship rule; that is, the attributes that will be evaluated by the survivorship rule to determine the most current data.

You can select preconfigured attributes groups or define unique attribute groups. For information on creating attribute groups, see “Creating Attribute Groups for Survivorship Rules” on page 84.

7 Enter values for the columns in the record. See the following table for details on the available columns.

Column Name	Description
Comparison Rule	Select either History, Recent, or Source as the rule for comparison. Based on rule selected, UCM evaluates whether the input from external system can be used to update the specified attribute group values. For more information on each comparison method, see "About UCM Survivorship Rules" on page 82.
Sequence	The order in which to evaluate multiple attribute groups.

8 Refresh the GUI view by stepping off the record.

9 If comparison rule is by source for the attribute group, set the source confidence level as follows:

a Select the Source Confidence tab.

b In the System ID list, click New.

c In the new record, define the source system and set a confidence level for the individual source system.

d Add other systems as necessary.

10 Add multiple attribute groups, as in the previous step, to further define the survivorship rule.

Activating UCM Survivorship Rules

Activating UCM survivorship rules is a task in the "Process for Creating Survivorship Rules" on page 83. For further information on survivorship rules, see "About UCM Survivorship Rules" on page 82.

NOTE: Review UCM survivorship rules thoroughly in a test environment before activating the rules.

To activate UCM Survivorship Rules

1 From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.

2 From the link bar, select Survivorship Rules

3 Select the survivorship rule of interest.

4 Select the Summary tab.

5 Review the information contained in this view for the survivorship rule.

6 Select the Attribute Groups tab.

7 Select the survivorship rule of interest and set the status field to Active.

Example of a Survivorship Rule

This topic describes the default Survivorship rules included in the Universal Customer Master (UCM), and provides an example of a default rule with modifications. You may use this feature differently, depending on your business model.

The Universal Customer Master (UCM) includes two default rules named Contact Default and Account Default. There are no explicitly defined attribute groups for either of these rules in the attribute group list in the lower view, see Figure 11. This status implies that the default attribute groups—those visible in the Attribute Groups list after selecting the Attribute Group Fields tab—are evaluated by the default criteria field in the default survivorship rule.

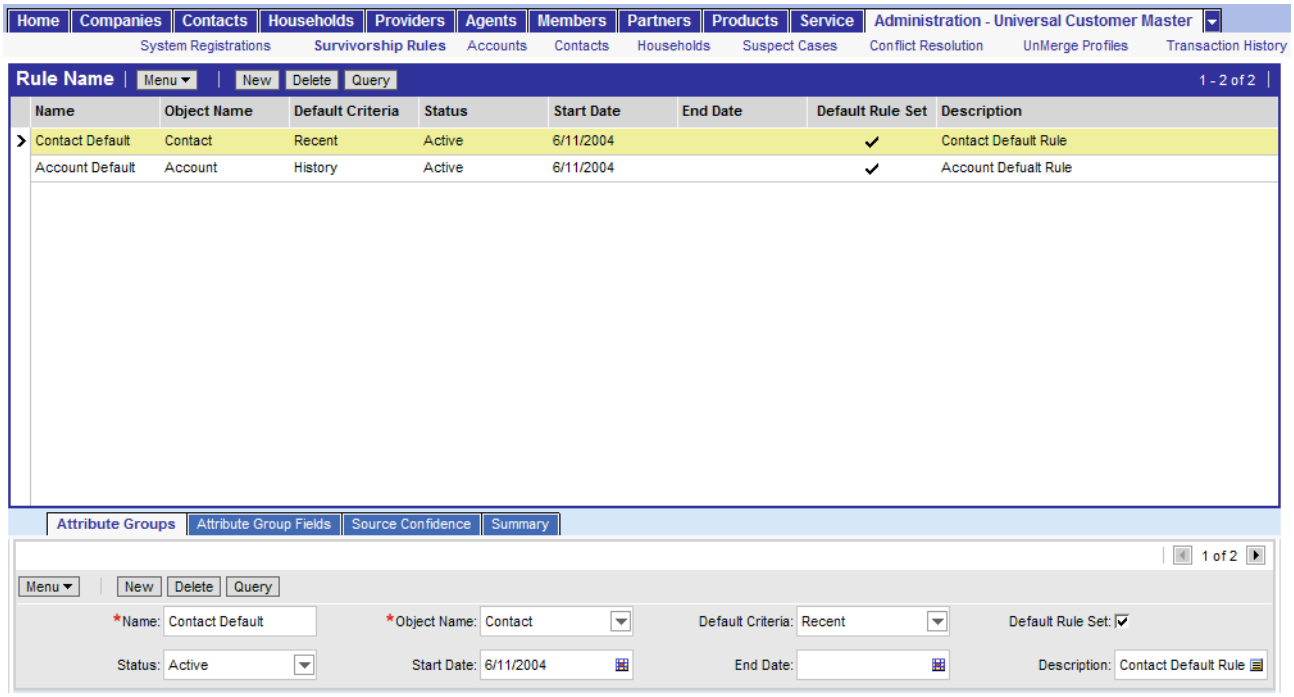

Figure 11. Default Contact and Account Survivorship Rules

The Status field for both rules is Active and the Start Date field for both rules is June 11, 2004, which indicates that this rule is ready to evaluate updates to the UCM for the default attribute groups and their associated fields.

The default criteria field for the Contacts Default rule is recent. This default criteria indicates that the Contacts Default rule evaluates the *date* of the incoming external system update against the *date* of the existing data saved in the default attribute groups. The most recent date of the two data records survives in the UCM record.

The default criteria field for the Accounts Default rule is history. This default criteria indicates that the Accounts Default rule evaluates the same incoming updates by date; however, the older data survives in the UCM record.

The following example describes creating a modification to the Contacts Default rule to enhance the survivorship rule by comparing updates to a Contact's income data. Start by creating a new attribute group:

To create a new attribute group for the Contacts Default rule

1. From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.
2. From the link bar, select Survivorship Rules, and select the Attribute Group Fields tab.
3. In the Attribute Group list, click the New button.
4. Define the new attribute group as follows:

   Attribute Group Name: Income Attribute Group
   Object: Contact
   Description: Defines rule for income field in UCM

5. In the Attribute Group Field list, click the New button:
6. Add the field name Income to the new attribute group from the drop-down menu.

The new attribute group is now defined and references the Income field. Next, attach the new attribute group to the Contacts Default rule and define the comparison method as by source:

To attach the new attribute group to the Contacts Default rule

1. From the link bar, select Survivorship Rules.
2. Set the Status field to inactive while defining the modification to the default rule.
3. In the Survivorship Rules list, select the Contact Default rule.
4. In the Attribute Group list, add a new attribute group by clicking New.
5. From the Attribute Group picklist, query for and select the attribute group created previously, Income Attribute Group.
6. Define the Comparison Rule for this attribute group as By Source.
7. Step off or save the attribute group record for the Contacts Default rule.
8. Because the comparison method is By Source, define the confidence level of the source external systems for the new attribute group:
   a. Select the Source Confidence tab.
   b. In the System ID list, click New.
   c. In the new record, define the source system. As an example, let Source A be the name of one external system and set this system's confidence level to 75.
   d. Add another source record. As an example, let Source B be the name of a second external system and set this system's confidence level to 95.

This survivorship rule evaluates all updates to the Income field based on the confidence level of the source. For example, if Source B updates the field originally, and then Source A updates at a later date, Source B's data survives because the confidence level is greater than Source A's. Alternatively, if the comparison method was Recent, Source A's data for the Income field would update the UCM record and replace Source B's data.

The final task in the process is to review and activate the enhanced rule:

To review and activate the enhanced Default Contact rule

1. From the link bar, select Survivorship Rules
2. Select the Contact Default survivorship rule.
3. Select the Summary tab, and review the details of the modified Contact Default rule.
4. Navigate back to the survivorship rule list, and select the Contact Default rule.
5. Set the Status field to Active, and make sure the Start and End date fields do not invalidate this rule.

The modified Contact Default rule is now active and evaluates updates to the Income field.

About Managing UCM Data

Managing data stored in the Universal Customer Master (UCM) is necessary to make sure that data is accurate and up-to-date. Several screens and views are available to the administrator to monitor the history of UCM data and the source of any updates. Administrators can also manually merge duplicate records identified by UCM or roll-back updates that were previously merged. For information on the these and other management tasks, see the following sections:

- "Cross-Referencing Records with External Systems" on page 90
- "About SDH Tables" on page 91
- "Monitoring Source Data of UCM Records" on page 92
- "Purging Source UCM Data" on page 92
- "Managing Pending Updates from External Systems" on page 93
- "Managing Duplicate UCM Records" on page 94
- "Unmerging Previously Merged UCM Records" on page 94

About Cross-Referencing Records with External Systems

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 more records in each registered application 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 UCM. The Unique Customer Identification number from external systems is embedded within the <Id> tag of the request message.

The System Administrator can manually cross-reference records from the Siebel Master Data Applications administration Screen. For information on this task, see "Cross-Referencing Records with External Systems" on page 90.

Cross-Referencing Records with External Systems

Cross-reference Universal Customer Master (UCM) records with external systems to store source identification data in the UCM. For further information on cross-referencing records, see "About Cross-Referencing Records with External Systems" on page 90.

For further information on managing data in the UCM, see "About Managing UCM Data" on page 89.

To cross-reference a contact, account, or household record

1. From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.
2. From the link bar, select either Contacts, Accounts, or Households.
3. In the Contacts, Accounts, or Households list, select the record of interest.
4. Select the External IDs tab for Contacts, Accounts, or Households.
5. In the External Account IDs view, click New.

6 Enter the appropriate information in the available fields to define the external customer and the Unique Account Identification, for each Contact, Account, or Household party record in Siebel UCM.

Use Table 18 for information on each of the fields.

Table 18. Reference Record Fields

Field	Description
System ID	The System Id represents an identifier for an application that accesses Siebel Master Data Applications. Pick this number 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.

About SDH Tables

The UCM Source Data History tables (SDH tables) contain the transactional contact and account data records pertaining to Siebel UCM and registered external systems. These tables hold incoming, best version, and historical data records, and provide the content for the UCM administration views. The storage of this data allows for Siebel UCM features such as survivorship, merge and unmerge, and so on.

The SDH tables are named with the S_UCM_* prefix such as S_UCM_CONTACT and S_UCM_ORG_EXT. The UCM base tables are the tables that are visible from the Siebel application such as S_PARTY, S_CONTACT, S_USER, and stores the master copy of the record.

The SDH tables will ultimately contain a large amount of data and should be purged at regular intervals. For further information on this task, see “Purging Source UCM Data” on page 92.

Monitoring History of UCM Records

Monitoring the history of UCM records reveals how the data stored in the UCM evolved and from what external system or systems the UCM record takes its content.

For information on other UCM data management tasks, see “About Managing UCM Data” on page 89.

To monitor history of UCM records

1. From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.
2. From the link bar, select either Contacts or Accounts.
3. In the Contacts or Accounts list, select the record of interest.
4. Select the Historical Version tab.
5. In this list, review the records from different source systems that make up the UCM version of the record in the top form. This list also contains historical versions of that particular UCM record within UCM.

Monitoring Source Data of UCM Records

Monitoring the source data and history of UCM records from this view reveals a global view of the evolution of UCM records.

For information on other UCM data management tasks, see "About Managing UCM Data" on page 89.

To monitor source data for Contacts or Accounts

1. From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.
2. From the link bar, select either Transaction History.
3. Select either Account Source Data or Contact Source Data.
4. In this list, review the records that show the source data and history records for contacts or accounts (content is similar to the Historical Version view except that the Historical Version view only shows source and history data for a specific UCM record; this view shows source data and history data for all UCM records).

Purging Source UCM Data

Maintaining historical source data for Universal Customer Master (UCM) records leads to large amounts of information stored in the database (SDH Tables). UCM provides a purge functionality to clear this data when necessary.

For information on other UCM data management tasks, see "About Managing UCM Data" on page 89.

CAUTION: The purge data operation removes all records and cannot be undone.

To purge source data

1. From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.
2. From the link bar, select either Transaction History.

3 Select either Account Source Data or Contact Source Data.

4 In this list view, click Purge.

Managing Pending Updates from External Systems

Duplicate Universal Customer Master (UCM) records are identified by the application and listed in the Suspect Cases screen. This screen displays incoming record updates from external systems that match to:

- Multiple existing records above the automatic threshold
- No existing records above automatic threshold but one or multiple existing records between manual threshold and automatic threshold

Administrative users can evaluate the records and decide whether to update the records manually or create new records to add to the UCM.

When updating records—by selecting the Link and Update button—the UCM application:

- Saves a copy of the existing best version record into the record's History.
- Links the record from the external system to the existing best version in UCM.
- Invokes survivorship rules to produce a new best version of the record.

For information on other UCM data management tasks, see "About Managing UCM Data" on page 89.

To update or promote duplicate UCM records

1 From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.

2 From the link bar, select Suspect Cases.

3 Select either the Duplicate Accounts tab or Duplicate Contacts tab.

4 In the Account or Contact Source list, select the source record of interest.

The records in this list are identified as duplicates of the base UCM record in the lower view. The record columns provide detail on each Account or Contact, including the source external system where the record originated.

5 In the lower Duplicate tab, review the UCM record that is in conflict with the record or records in the Source list.

The record columns in this list provide details on the existing Account or Contact in the UCM. The score field determines how close the incoming record matches the existing record.

6 After review, select the source record of interest and update the two records by clicking Link and Update or promote the new record by clicking Create New.

Managing Duplicate UCM Records

The Universal Customer Master (UCM) identifies and lists duplicate Contact and Account records for review by administrators after running data quality data matching in batch mode. The administrator can resolve the duplicate records with one of the following tasks:

- Merging the duplicate record with the existing UCM record. The merging process:
  - Stores the duplicate record as type Merged and the surviving record as History in the Source Data History table
  - Merges the parent records using survivorship rules to generate the new merged surviving parent record
  - Updates the parent surviving record with the new merged surviving record
  - Adds children records from duplicate record to the surviving record.
  - Deletes the duplicate record.
- Deleting the duplicate record and leaving the existing UCM record unchanged.

To review and manage duplicate UCM records

1. From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.
2. From the link bar, select Conflict Resolution.
3. Select the Duplicate Contacts or Duplicate Accounts tab.

   Review the duplicate contacts or accounts in this view.
4. Review an individual duplicate record by drilling down on the record's hyperlink.

   The Duplicate Account Resolution or Duplicate Contact Resolution view appears. The top record is the current UCM version of the account or contact. The bottom view is the duplicate record.
5. Resolve the duplicate record by:
   a. Merging the duplicate record with the UCM record, by clicking the Merge button.
   b. Deleting the duplicate record by clicking the Menu button and then Delete Record.
   c. Deleting the UCM record by clicking the Delete button.

Unmerging Previously Merged UCM Records

A Siebel Universal Customer Master (Siebel UCM) record that results from the merging of two preexisting records, as defined by survivorship rules, can be unmerged to the state of their original records. After unmerging the merged record, Siebel UCM restores the parent victim record and reverts the parent merged record back to its previous state. Siebel UCM removes children of the victim record from the surviving merged record and adds them back to the victim record. (Siebel UCM supports a limited number of children.)

If the merged record is subsequently updated prior to rolling back, the surviving unmerged record retains the updates, while the record that did not survive during the original merge reverts to its original state.

To unmerge a merged UCM record

1. From the application-level menu, choose Navigate > Site Map > Administration - Universal Customer Master screen.
2. From the link bar, select Unmerge Profiles.
3. Select either the Merged Accounts tab or Merged Contacts tab.
4. In the Account or Contact Source list, select the source record of interest.
5. Click the Unmerge button.

Running Data Management in Batch Mode

To run Siebel Universal Customer Master (Siebel UCM) data management processes in batch mode, you create a component job based on the server component UCM Batch Manager (alias: UCMBatchProcess).

This component job invokes the batch workflow. The workflow queries, cleanses, matches, and publishes the UCM records. The batch process performs matching of the incoming record against the existing records in Siebel UCM. There are three match cases that can occur: no match, one match, or multiple matches. The workflow internally calls the transaction manager and survivorship rules, depending upon the match case type.

NOTE: If there are duplicates within the same batch transaction, Siebel UCM captures the duplicates and displays them in the Conflict Resolution view where data stewards can review and merge the records as appropriate.

For background information on administering component jobs, see *Siebel System Administration Guide*.

To run UCM data management in batch mode by scheduling a batch workflow

1. From the application-level menu, choose Navigate > Site Map > Administration - Server Management.
2. From the link bar, click Jobs.
3. In the Jobs list, click New.

   The component job status field changes to Creating.
4. In the Component/Job field, click the drop-down list and select UCM Batch Manager.
5. In the Job Detail view, enter data in appropriate fields to describe the start time and so on of the component job; click the menu button, and then click Save Record.

6 In the Job Parameters list, add the following parameters for the component job:

Parameter Name	Value
UCM Batch Object Type	[ObjectType; SearchSpec; SortSpec] For example: `Contact;[Contact.UCM Type Code] = 'Batch';Created (ASCENDING), Last Name, First Name`
UCM Batch Size	Select the page size of the batch job. The default value is 10.
UCM Data Management Flag	Make sure this value is set to TRUE.
UCM Survivorship Engine Flag	Make sure this value is set to TRUE.

7 In the Jobs list, click the Start button.

The Status field changes from Creating to Queued.

About System Preferences for Siebel UCM

Siebel UCM system preferences control how the application operates and what functionality is enabled in your environment. You may need to review default settings and modify these as appropriate based on your business needs. For further information on setting system preferences, see *Applications Administration Guide*.

Table 19 lists and briefly describes the Siebel UCM system preferences.

Table 19. Siebel UCM System Preferences

System Preference Name	Default Value	Description
Enable CDM Cleanse: Account	FALSE	Turn on CDM Cleanse for Accounts
Enable CDM Cleanse: Contact	FALSE	Turn on CDM Cleanse for Contacts
Enable CDM EM: Account	FALSE	Turn on CDM Exact Match for Accounts
Enable CDM EM: Contact	FALSE	Turn on CDM Exact Match for Contacts
Enable CDM Match: Account	FALSE	Turn on CDM Match for Accounts
Enable CDM Match: Contact	FALSE	Turn on CDM Match for Contacts
Enable Survivorship Engine	FALSE	Turn on Survivorship Engine

Table 19. Siebel UCM System Preferences

System Preference Name	Default Value	Description
Enable UCM Processes	FALSE	Set this value to TRUE to enable integration with other applications and Siebel UCM. For further information, see "Configuring Deep Copy for Data Hierarchies in UCM" on page 65.
UID Generation Service	"FINS Teller Converter Extensions, ""GenerateID"", ""IDType|GUID"" "	The business service name, method, and input combination to generate unique ID value. It is used by both transaction manager and UI Interaction service.
Unique Id: Account	Party UId	Unique ID field for Account is used with transaction manager and UI Interaction
Unique Id: Contact	Party UId	Unique ID field for Contact is used with transaction manager and UI Interaction
Unique Id: FINCORP Account	Account Number	Unique ID field for FINCORP Account is used with transaction manager and UI Interaction
Unique Id: Household	Party UId	Unique ID field for Household is used with transaction manager and UI Interaction

Configuring Siebel Data Quality Cleansing for Siebel UCM

Siebel Universal Customer Master (Siebel UCM) supports data cleansing using the Siebel Data Quality module. To configure cleansing of Siebel UCM data, perform the following task.

NOTE: Siebel Data Quality module is licensed separately from Siebel UCM.

For further information on Siebel Data Quality, see *Siebel Data Quality Administration Guide*.

NOTE: The following task contains references to many system administration tasks. For further detail and background information on system administration tasks, see *Siebel System Administration Guide*.

To configure Siebel Data Quality cleansing for Siebel UCM

1 Log in to Siebel Business Applications. You must have appropriate system administration responsibilities to make the following configurations.

2 From the application-level menu, choose Navigate > Site Map > Administration - Server Configuration > Enterprises > Parameters to make sure the following enterprise parameters are set:

Parameter	Value
UCM Data Management Flag	TRUE
Data Cleansing Type	Vendor1

3 From the application-level menu, choose Navigate > Site Map > Administration - Application screen > System Preferences view, and set the following system preferences to TRUE (the default values are FALSE):

System Preference	Value
Enable CDM Cleanse: Account	TRUE
Enable CDM Cleanse: Contact	TRUE

For background information on setting system preferences, see *Applications Administration Guide*.

4 From the application-level menu, choose Navigate > Site Map > Administration - Server Configuration, and click the Component Definitions view tab.

5 For each of the following component definitions:

- Data Quality Manager (alias Dqmgr)
- UCM Object Manager (alias UCMObjMg)
- UCM Batch Manager (alias UCMBatchProcess)
- UCM Batch Publish Subscribe (alias UCMBatchPubSub)
- Workflow Process Batch Manager (alias WfProcBatchMgr)
- Workflow Process Manager (alias WfProcMgr)

set the following parameters:

Parameter	Value
Data Cleansing Enable Flag	TRUE
Data Cleansing Type	Vendor1

6 From the application-level menu, choose Navigate > Site Map > Administration - Server Configuration > Enterprises > Synchronize, and click the Synchronize button to synchronize the changes made to the batch-mode server components.

7 From the application-level menu, choose Navigate > Site Map > Administration - Data Quality > Data Quality Settings, and set the following parameters:

Name	Value
Disable Data Cleansing	No
Key Type	Standard
Match Threshold	90
Search Type	Typical

8 From the application-level menu, choose Navigate > Site Map > User Profile Preferences > Data Quality, and set the following parameters:

Parameter	Value
Disable Data Cleansing	No

9 Configure the UCM configuration file (ucm.cfg) by setting the following parameters in the `[DataCleansing]` section:

- `Data Cleansing Enable = TRUE`
- `Data Cleansing Type = Vendor1`

10 Stop and restart the Siebel Server and Siebel Gateway Name Server system services for changes to take effect.

Configuring SSA Data Matching for Siebel UCM

Siebel Universal Customer Master (Siebel UCM) support data matching using SSA-NAME3 (SSA) third-party software. To configure this functionality for Siebel UCM, perform the following task.

For further information on SSA configurations, see *Siebel Data Quality Administration Guide*.

NOTE: The following task contains references to many system administration tasks. For further detail and background information on system administration tasks, see *Siebel System Administration Guide*.

To configure SSA data matching for Siebel UCM

1 Log in to Siebel Business Applications. You must have appropriate system administration responsibilities to make the following configurations.

2 From the application-level menu, choose Navigate > Site Map > Administration - Server Configuration > Enterprises > Parameters to make sure the following enterprise parameters are set:

Parameter	Value
UCM Data Management Flag	TRUE
DeDuplication Data Type	SSA

3 From the application-level menu, choose Navigate > Site Map > Administration - Application screen > System Preferences view, and set the following system preferences to TRUE (the default values are FALSE):

System Preference	Value
Enable CDM EM: Account	TRUE
Enable CDM EM: Contact	TRUE
Enable CDM Match: Account	TRUE
Enable CDM Match: Contact	TRUE

For background information on setting system preferences, see *Applications Administration Guide*.

4 From the application-level menu, choose Navigate > Site Map > Administration - Server Configuration, and click the Component Definitions view tab.

5 For each of the following component definitions:

- Data Quality Manager (alias Dqmgr)
- UCM Object Manager (alias UCMObjMg)
- UCM Batch Manager (alias UCMBatchProcess)
- UCM Batch Publish Subscribe (alias UCMBatchPubSub)
- Workflow Process Batch Manager (alias WfProcBatchMgr)
- Workflow Process Manager (alias WfProcMgr)

set the following parameters:

Parameter	Value
DeDuplication Data Type	SSA
DeDuplication Enable Flag	TRUE

6 From the application-level menu, choose Navigate > Site Map > Administration - Server Configuration > Enterprises > Synchronize, and click the Synchronize button to synchronize the changes made to the batch-mode server components.

7 From the application-level menu, choose Navigate > Site Map > Administration - Data Quality > Data Quality Settings, and set the following parameters:

Name	Value
Disable Deduplication	No
Key Type	Standard
Match Threshold	90
Search Type	Typical

8 From the application-level menu, choose Navigate > Site Map > User Profile Preferences > Data Quality, and set the following parameters:

Parameter	Value
Disable Deduplication	No

9 Configure the UCM configuration file (ucm.cfg) by setting the following parameter in the `[DeDuplication]` section:

- `DeDuplication Enable = TRUE`

10 Stop and restart the Siebel Server and Siebel Gateway Name Server system services for changes to take effect.

About Siebel UCM Data Hierarchy

Siebel UCM supports the use of account data hierarchies with the master customer data, and includes screens to display and view the account hierarchies. Siebel UCM also supports copying existing hierarchies with their existing relationships.

For further information on copying data hierarchies, see:

- "Copying Siebel UCM Data Hierarchies" on page 102

The following tables allow for this functionality: S_DYN_HRCHY, which stores hierarchy information as a header; S_DYN_HRCHY_REL, which stores relationships; and S_DYNHR_RPT_REL, which stores denormalized hierarchy relationships.

The data hierarchy information is available for viewing at the UCM Hierarchy Admin screen.

For further background information on data hierarchy concepts, see *Applications Administration Guide*.

Copying Siebel UCM Data Hierarchies

This topic describes the task of a Siebel UCM data hierarchies deep copy. A deep copy of a data hierarchy copies not only the hierarchy header but all the relationships in the hierarchy as well. For further information on Siebel UCM data hierarchies, see "About Siebel UCM Data Hierarchy" on page 101.

To run deep copy of a Siebel UCM data hierarchies

1. Navigate to the UCM Hierarchy Admin screen.
2. Select the data hierarchy of interest.
3. From the menu, select Copy Record to run the deep copy.

Troubleshooting Siebel Master Data Applications

This topic provides guidelines for resolving problems with Siebel Master Data applications.

To resolve the problem, look for it in the list of error messages in Table 20.

Table 20. Resolving Siebel Master Data Application Error Messages

Error Number	Error Text	Explanation	Resolution
SBL-IAI-00429	%1 Checking system registration for system %2	Checking if system is registered in UCM.	Informational: no action required.
SBL-IAI-00427	%1 Checking if privilege is allowed for %2 operation	Checking if system has privilege to perform the operation.	Informational: no action required.
SBL-IAI-00428	%1 Checking if security is enforced	Checking if security is enabled in UCM.	Informational: no action required.
SBL-IAI-00430	%1 Generating Unique Id for %2	UCM is generating a new Unique ID for this message.	Informational: no action required.
SBL-IAI-00431	%1 Inserting reference record for %2 with Id %3	Inserting External ID record.	Informational: no action required.

Table 20. Resolving Siebel Master Data Application Error Messages

Error Number	Error Text	Explanation	Resolution
SBL-IAI-00432	%1 Fail to find %2	A service corresponding to %1 fail to find object corresponding to %2.	Check to make sure indicated object exists.
SBL-IAI-00433	%1 Fail to find integration object instance	A service corresponding to %1 fail to find integration object instance.	This error indicates that no data or no valid data was received. Check that an integration object instance exists.
SBL-IAI-00434	%1 Fail to find Integration Object	A service corresponding to %1 fails to find the integration object name.	Check that an integration object instance exists in the property set.
SBL-IAI-00435	%1 Fail to find message instance under XML Hierarchy	Not able to find message in input XML.	Check that the XML document sent to the UCM contains a message body.
SBL-IAI-00436	%1 Fail to find System Id	No system number matches the system ID in input XML.	The System ID value is missing in the message. Provide a valid System ID in the input XML.
SBL-IAI-00437	%1 Failed to find XML Hierarchy	Input XML message does not contain valid XML.	Check if the input property set contains a child XML hierarchy property set.
SBL-IAI-00438	%1 Fail to find privilege record for %2 object	Required privilege is not allowed on the object.	Check in the UCM administration views that the system is registered and has privileges defined for it. For information on this task, see "Registering Systems Connected to Siebel Master Data Applications" on page 73 and "Setting System Privileges for Siebel Master Data Applications" on page 76.
SBL-IAI-00439	%1 Fail to find system registration	Unable to find required system in system registration view.	Check that the incoming message has a valid System ID by verifying that the SystemId field value is not null, and that the SystemId is registered in the UCM administration views.
SBL-IAI-00440	%1 Updating reference record for %2 with Id %3	Updating external ID for the record.	Informational: no action required.

Table 20. Resolving Siebel Master Data Application Error Messages

Error Number	Error Text	Explanation	Resolution
SBL-IAI-00443	%1 Found no match	No match found.	Informational: no action required
SBL-IAI-00444	%1 Found multiple instances within the input	Multiple accounts or contacts found in the input from SDH record. Currently CDM supports only one record.	Check the input data and verify that only one instance (for example, of an account or contact) is supplied.
SBL-IAI-00467	%1 Found multiple selections from UI	Multiple selection from the dedup UI.	Select only one row in the applet when running Link and Update.
SBL-IAI-00447	%1 Found no data cleansing output	No data cleansing output found. Data quality cleanse did not return any valid output.	Check the data cleansing service and parameters to verify it has been configured correctly.
SBL-IAI-00445	%1 Found no instance within input	No instance found in the input data.	Verify that the input data is not null, and the data is in a valid format.
SBL-IAI-00465	%1 Found no user property key definition for %2 in dispatch map	Cannot find the user property key in the dispatch map integration object definition.	Check that there is a user property defined in the UCM Dispatch Map integration object, that is, CIFDispMap.
SBL-IAI-00446	%1 has no manual threshold definition in user property	No manual threshold definition found in the user property of CDM service.	Check if a manual threshold definition exists in UCM Data Quality Manager Business Service user properties, and make sure the value is between 0 to100.

Table 20. Resolving Siebel Master Data Application Error Messages

Error Number	Error Text	Explanation	Resolution
SBL-IAI-00449	%1 Search fail to return results from %2	The search failed to return active row back.	The search failed to return an active row and one active row is expected. You may receive an error in the following cases: ■ When updating the Duplicate Flag in UCM table, UCM does a query for the duplicate record and does not find it. ■ When querying for the cleansed version of source data, UCM does a query for the cleansed record and does not find it. ■ When querying the business component within the Deduplication business object, UCM does a query for the field used in deduplication, and does not find the field. ■ When cleaning up the selected data in the business component within the Deduplication business object, UCM does a query for the business component record and does not find a record. ■ When clearing the UCM source table Duplicate Flag, UCM does a query for the duplicate record and does not find it.
SBL-IAI-00448	%1 PropertySet doesn't match the type %2	The type for the property set is not correct. Type=Match Info. Data quality match returns property set of type Match Info.	Check the output property set of your Deduplication business service. The property set should have a Type of Match Info.
SBL-IAI-00451	%1 Fail to find %2 with %3	A service corresponding to %1 fail to find object corresponding to %2 with ID corresponding to %3.	Check to make sure an object exists for the ID supplied.

Table 20. Resolving Siebel Master Data Application Error Messages

Error Number	Error Text	Explanation	Resolution
SBL-IAI-00474	%1 Child record %2 not found	Reparenting of one of the child records has failed because child record ID is not found.	The selected record could not be unmerged. This may have been due to the other child record being deleted.
SBL-IAI-00475	%1 Failed to reparent child %2	Reparenting of one of the child records fails because child record ID is not found.	Informational: no action required.
SBL-IAI-00461	%1 Invalid Match Id %2 for AutoMatch	A service corresponding to %1 find invalid Match ID %2 for AutoMatch.	Check to make sure there is a valid Match ID value supplied for the AutoMatch function and that it is not null.
SBL-IAI-00458	%1 Invalid value %2 for parameter Batch Size	A service corresponding to %1 find invalid value %2 for input parameter Batch Size.	Check to make sure the input argument for the Batch Size is an integer and greater than zero.
SBL-IAI-00459	%1 Invalid value %2 for parameter Sleep Time	A service corresponding to %1 find invalid value %2 for input parameter Sleep Time.	Check to make sure the input argument Sleep Time is an integer and equal to or greater than zero.
SBL-IAI-00460	%1 Fail to find parameter Object Type	A service corresponding to %1 fail to find required input parameter Object Type.	Check to make sure the input argument has an Object Type. The Object Type choices are Contact, Account, or Household.
SBL-IAI-00462	%1 Fail to query current Best Version with %2	Unable to query best version record from base table while preparing history record. This scenario could exist because no record is satisfying the user key in IO (that is, like fields not present in input, or fields are empty in input).	Check to make sure the incoming record contains either a Party_UID or Row_Id, or both. If any of these ID values are supplied, check that there is only one matching record in the SDH tables with that value.

Table 20. Resolving Siebel Master Data Application Error Messages

Error Number	Error Text	Explanation	Resolution
SBL-IAI-00463	%1 Fail to update Source Record %2	After the operation, when linking SDH record to the best version record, unable to update best version ID in SDH record in SDH table. Reason could be because no user key fields are in input.	Informational: no action required.
SBL-IAI-50102	%1 Creating Default Attribute Group Data	Creating default attribute group data for the new object instance.	Informational: no action required.
SBL-IAI-50100	%1 Getting Attribute Group Data	Retrieving attribute group data for the object.	Informational: no action required.
SBL-IAI-50101	%1 Getting survivorship rule	Retrieving active or default Survivorship Rule.	Informational: no action required.
SBL-IAI-50103	%1 Inserting Attribute Group Data	Inserting attribute group data for the corresponding object.	Informational: no action required.
SBL-IAI-00455	%1 Fail to find Rule Set	A service corresponding to %1 fails to find configured default Rule Set.	Check to make sure a default and nonexpired Rule Set for the Object is configured in UCM administration for survivorship. For further information on configuring survivorship, see "Process for Creating Survivorship Rules" on page 83.
SBL-IAI-00457	%1 Fail to find Default Attribute Group	A service corresponding to %1 fails to find configured default attribute group.	Check to make sure a default and nonexpired Attribute Group for the object is configured in UCM Administration for survivorship. For further information on configuring survivorship, see "Process for Creating Survivorship Rules" on page 83.
SBL-IAI-50104	%1 Processing base fields	Processing and intelligently merging the supported base fields in the object.	Informational: no action required.

Table 20. Resolving Siebel Master Data Application Error Messages

Error Number	Error Text	Explanation	Resolution
SBL-IAI-00476	%1 Unmerge parent record %2 failed	Either the Unmerge parent record fails or when unmerge is done after merge operation (without any update operation), then surviving record is refreshed. If this refresh fails then this error occurs.	Informational: no action required.

Table 20. Resolving Siebel Master Data Application Error Messages

Error Number	Error Text	Explanation	Resolution
SBL-IAI-00477	%1 Clean last unmerged details %2 failed	After unmerge process is completed, either deletion of child records from merge table fails or setting the state of the victim record from merged to unmerged, fails.	Either one of the following cases occurs: ■ The unmerge process completed successfully but the child records were not deleted. Review the application logs for detailed messages on why the child records were not deleted. Most likely there is a database issue you will need to resolve. You will need to work with your database administrator to manually delete the child records from the database. ■ The unmerge process completed successfully but the victim record status was not changed from merged to unmerged. Review the application logs for detailed messages on why the status could not be updated. Most likely there is a database issue you will need to resolve. You will need to work with your database administrator to manually update the status value in the database.
SBL-IAI-00476	%1 Unmerge parent record %2 failed	Either the Unmerge parent record fails or when unmerge is done after merge operation (without any update operation), then surviving record is refreshed. If this refresh fails then this error occurs.	Informational: no action required.

8 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 21 lists each of the messages from Siebel Master Data Applications specifications.

Table 21. 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 21. 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 21. 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
CIFAccountUpsert	UpsertAccount
CIFAccountUpsertRs	UpsertAccountResponse
CIFContact	LookUpContact
CIFContactRs	LookUpContactResponse
CIFContactActivity	LookUpContactActivity
CIFContactActivityRs	LookUpContactActivityResponse
CIFContactActivityByName	LookUpContactActivityByName
CIFContactActivityByNameRs	LookUpContactActivityByNameResponse
CIFContactActivityDelete	DeleteContactActivity
CIFContactActivityDeleteRs	DeleteContactActivityResponse
CIFContactActivityInsert	InsertContactActivity

Table 21. Siebel Universal Customer Master Application Messages

Integration Object Name	Business Processes / Operations
CIFContactActivityInsertRs	InsertContactActivityResponse
CIFContactActivityUpdate	UpdateContactActivity
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

Table 21. Siebel Universal Customer Master Application Messages

Integration Object Name	Business Processes/Operations
CIFContactProfile	LookUpContactProfile
CIFContactProfileRs	LookUpContactProfileResponse
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
CIFContactUpsert	UpsertContact
CIFContactUpsertRs	UpsertContactResponse
CIFFINCORPAccount	LookUpFINCORPAccount
CIFFINCORPAccountRs	LookUpFINCORPAccountResponse
CIFFINCORPAccountByName	LookUpFINCORPAccountByName
CIFFINCORPAccountByNameRs	LookUpFINCORPAccountByNameResponse
CIFHousehold	LookUpHousehold
CIFHouseholdRs	LookUpHouseholdResponse
CIFHouseholdByName	LookUpHouseholdByName

Table 21. Siebel Universal Customer Master Application Messages

Integration Object Name	Business Processes/Operations
CIFHouseholdByNameRs	LookUpHouseholdByNameResponse
CIFHouseholdDelete	DeleteHousehold
CIFHouseholdDeleteRs	DeleteHouseholdResponse
CIFHouseholdInsert	InsertHousehold
CIFHouseholdInsertRs	InsertHouseholdResponse
CIFHouseholdUpdate	UpdateHousehold
CIFHouseholdUpdateRs	UpdateHouseholdResponse
CIFPersonalAddress	LookUpPersonalAddress
CIFPersonalAddressRs	LookUpPersonalAddressResponse
CIFPersonalAddressByState	LookUpPersonalAddressByState
CIFPersonalAddressByStateRs	LookUpPersonalAddressByStateResponse

About CRMML Messages

CRMML messages (Customer Relationship Management Markup Language) are a Siebel XML specification for customer-centric data exchange with other enterprise applications and Siebel Universal Customer Master (Siebel UCM) applications. External applications send CRMML request messages to Siebel applications and receive CRMML messages in response.

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>

      <PartyPackage>
         <InsertContactProfile>
               <Contact>
                     <Id>ext2223</Id>
                     <EmailAddress>cifemail</EmailAddress>
                     <FaxPhone>5555555555</FaxPhone>
                     <FirstName>0099INFN</FirstName>
                     <HomePhone>1111111111</HomePhone>
                     <LastName>0099INLN</LastName>
               </Contact>
         </InsertContactProfile>
      </PartyPackage>
</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>
<PartyPackage ID=”p1”>
    <InsertContactProfile ID=”m1”>
```

```
<Contact ID=”a1”>
      <Id>ext2223</Id>
```

The first <ObjectDescriptor> refers to <PartyPackage> element in the message body by HREF. It indicates that <PartyPackage> 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.

Certain values in the header are automatically generated by the connector, but you can specify others in the envelope integration object. Table 22 describes the elements and how they are configured. For system generated attributes, there is no configuration available.

NOTE: The CRMML envelope object information can be configured, contain custom fields, and works for both input and output formats. However, UCM business services refer to information in this object and look for specific XML values. Make sure specific fields exist in the business services before configuring the CRMML for your needs.

Table 22. Header Elements

Element	Structure	Attribute	Description	Default	Configuration
CRMML	(top level)				
CRMHeader	In CRMML		Header aggregate		
FromSystem	In CRMHeader		FromSystem aggregate		

Table 22. 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 22. 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

Body Section. The 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>

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

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

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 12 describes the structure of this message.

Figure 12. UCM Contact Profile Structure

Table 23. Contact Profile Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfile	Contact	LookUpContactProfile	UK1:PartyUId UK2: Id
CIFContactProfileRs	Contact	LookUpContactProfileResponse	UK1:PartyUId UK2: Id

Table 24. Contact Profile Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfileByName	Contact	LookUpContactProfileByName	UK1:PartyUId UK2: Id
CIFContactProfileByNameRs	Contact	LookUpContactProfileByNameResponse	UK1:PartyUId UK2: Id

Table 25. Contact Profile Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfileUpdate	Contact	UpdateContactProfile	UK1:PartyUId UK2: Id
CIFContactProfileUpdateRs	Contact	UpdateContactProfileResponse	UK1:PartyUId UK2: Id

Table 26. Contact Profile Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfileInsert	Contact	InsertContactProfile	UK1:PartyUId UK2: Id
CIFContactProfileUpdateRs	Contact	InsertContactProfileResponse	UK1:PartyUId UK2: Id

Table 27. Contact Profile Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProfileDelete	Contact	DeleteContactProfile	UK1:PartyUId UK2: Id
CIFContactProfileDeleteRs	Contact	DeleteContactProfileResponse	UK1:PartyUId UK2: Id

Table 28. Contact Profile Integration Component Fields

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

Table 28. Contact Profile Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
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
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		

Table 28. Contact Profile Integration Component Fields

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

Contact Address

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, InsertContactAddress, and DeleteContactAddress are message pairs for request and response operations. Figure 13 describes the structure of this message.

Figure 13. UCM Contact Address Structure

Table 29. Contact Address Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddress	Contact	LookUpContactAddress	UK1:PartyUId UK2: Id
CIFContactAddressRs	Contact	LookUpContactAddressResponse	UK1:PartyUId UK2: Id

Table 30. Contact Address Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddressByName	Contact	LookUpContactAddressByName	UK1:PartyUId UK2: Id
CIFContactAddressByNameRs	Contact	LookUpContactAddressByName Response	UK1:PartyUId UK2: Id

Table 31. Contact Address Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddressUpdate	Contact	UpdateContactAddress	UK1:PartyUId UK2: Id
CIFContactAddressUpdateRs	Contact	UpdateContactAddressResponse	UK1:PartyUId UK2: Id

Table 32. Contact Address Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddressInsert	Contact	InsertContactAddress	UK1:PartyUId UK2: Id
CIFContactAddressInsertRs	Contact	InsertContactAddressResponse	UK1:PartyUId UK2: Id

Table 33. Contact Address Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactAddressDelete	Contact	DeleteContactAddress	UK1:PartyUId UK2: Id
CIFContactAddressDeleteRs	Contact	DeleteContactAddressResponse	UK1:PartyUId UK2: Id

Table 34. 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	✔	

Table 34. Contact Address 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/M	MM	DTYPE_TEXT	15		Bound by MR_MS
Middle Name	MiddleName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100		
Person UId	PersonUId	DTYPE_TEXT	100		
Privacy Code	PrivacyCode	DTYPE_TEXT	30		
Status	Status	DTYPE_TEXT	30		Bound by CONTACT_STATUS
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		

Table 34. Contact Address Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
INS Personal Integration Id	INSPersonalIntegrationId	DTYPE_TEXT	30		
INS Personal Postal Code	INSPersonalPostalCode	DTYPE_TEXT	30		
INS Personal State	INSPersonalState	DTYPE_TEXT	10		
INS Personal Street Address	INSPersonalStreetAddress	DTYPE_TEXT	200	✔	
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 14 describes the structure of this message.

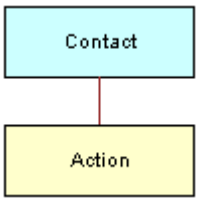

Figure 14. UCM Contact Activity Structure

Table 35. Contact Activity Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactActivity	Contact	LookUpContactActivity	UK1:PartyUId UK2: Id
CIFContactActivityRs	Contact	LookUpContactActivityResponse	UK1:PartyUId UK2: Id

Table 36. Contact Activity Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactActivityByName	Contact	LookUpContactActivityByName	UK1:Party UK2: Id
CIFContactActivityByNameRs	Contact	LookUpContactActivityByNameResponse	UK1:Party UK2: Id

Table 37. Contact Activity Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactActivityUpdate	Contact	UpdateContactActivity	UK1:PartyUId UK2: Id
CIFContactActivityUpdateRs	Contact	UpdateContactActivityResponse	UK1:PartyUId UK2: Id

Table 38. Contact Activity Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactActivityInsert	Contact	InsertContactActivity	UK1:PartyUId UK2: Id
CIFContactActivityInsertRs	Contact	InsertContactActivityResponse	UK1:PartyUId UK2: Id

Table 39. Contact Activity Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactActivityDelete	Contact	DeleteContactActivity	UK1:PartyUId UK2: Id
CIFContactActivityDeleteRs	Contact	DeleteContactActivityResponse	UK1:PartyUId UK2: Id

Table 40. 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		

Table 40. Contact Activity 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
Middle Name	MiddleName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100		
Person UId	PersonUId	DTYPE_TEXT	100		
Preferred Communications	PreferredCommunications	DTYPE_TEXT	30		Bound by OFFER_MEDIA
Status	Status	DTYPE_TEXT	30		Bound by CONTACT_STATUS
Work Phone #	WorkPhone	DTYPE_TEXT	40		
Work Phone Extension	WorkPhoneExtension	DTYPE_TEXT	10		
operation	operation	DTYPE_TEXT	30		
Action					
Account Id	AccountId	DTYPE_ID	15		
Account Location	AccountLocation	DTYPE_TEXT	50		

Table 40. Contact Activity Integration Component Fields

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

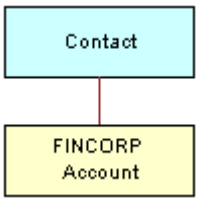

Figure 15. UCM Contact Product Structure

Table 41. Contact Product Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProduct	Contact	LookUpContactProduct	UK1:PartyUId UK2: Id
CIFContactProductRs	Contact	LookUpContactProductResponse	UK1:PartyUId UK2: Id

Table 42. Contact Product Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProductByName	Contact	LookUpContactProductByName	UK1:PartyUI UK2: Id
CIFContactProductByNameRs	Contact	LookUpContactProductByNameResponse	UK1:PartyUI UK2: Id

Table 43. Contact Product Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProductUpdate	Contact	UpdateContactProduct	UK1:PartyUId UK2: Id
CIFContactProductUpdateRs	Contact	UpdateContactProductResponse	UK1:PartyUId UK2: Id

Table 44. Contact Product Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProductInsert	Contact	InsertContactProduct	UK1:PartyUId UK2: Id
CIFContactProductInsertRs	Contact	InsertContactProductResponse	UK1:PartyUId UK2: Id

Table 45. Contact Product Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactProductDelete	Contact	DeleteContactProduct	UK1:PartyUId UK2: Id
CIFContactProductDeleteRs	Contact	DeleteContactProductResponse	UK1:PartyUId UK2: Id

Table 46. 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		

Table 46. Contact Product 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	255		Bound by SEX_MF
M/M	MM	DTYPE_TEXT	15		Bound by MR_MS
Middle Name	MiddleName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100		
Person UId	PersonUId	DTYPE_TEXT	100		
Preferred Communications	PreferredCommunications	DTYPE_TEXT	30		Bound by OFFER_MEDIA
Status	Status	DTYPE_TEXT	30		
Work Phone #	WorkPhone	DTYPE_TEXT	40		
Work Phone Extension	WorkPhoneExtension	DTYPE_TEXT	10		
operation	operation	DTYPE_TEXT	30		
FINCORP Account					
Account Branch ABA	AccountBranchABA	DTYPE_TEXT	50		Used for picking a related entity.
Account Branch Id	AccountBranchId	DTYPE_ID	15		

Table 46. Contact Product Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Account Holder	AccountHolder	DTYPE_TEXT	100		Used for picking a related entity.
Account Name	AccountName	DTYPE_TEXT	100		
Account Number	AccountNumber	DTYPE_TEXT	100	✔	
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		
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 16 describes the structure of this message.

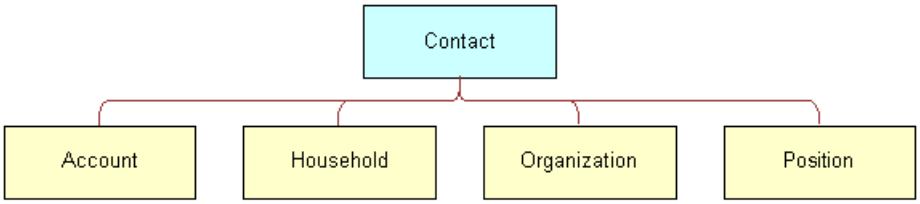

Figure 16. UCM Contact Relationship Structure

Table 47. Contact Relationship Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationship	Contact	LookUpContactRelationship	UK1:PartyUId UK2: Id
CIFContactRelationshipRs	Contact	LookUpContactRelationshipResponse	UK1:PartyUId UK2: Id

Table 48. Contact Relationship Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationshipByName	Contact	LookUpContactRelationshipByName	UK1:PartyUI UK2: Id
CIFContactRelationshipByNameRs	Contact	LookUpContactRelationshipByNameResponse	UK1:PartyUI UK2: Id

Table 49. Contact Relationship Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationshipUpdate	Contact	UpdateContactRelationship	UK1:PartyUI UK2: Id
CIFContactRelationshipUpdateRs	Contact	UpdateContactRelationshipResponse	UK1:PartyUI UK2: Id

Table 50. Contact Relationship Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationshipInsert	Contact	InsertContactRelationship	UK1:PartyUI UK2: Id
CIFContactRelationshipInsertRs	Contact	InsertContactRelationshipResponse	UK1:PartyUI UK2: Id

Table 51. Contact Relationship Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactRelationshipDelete	Contact	DeleteContactRelationship	UK1:PartyUI UK2: Id
CIFContactRelationshipDeleteRs	Contact	DeleteContactRelationshipResponse	UK1:PartyUI UK2: Id

Table 52. 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 52. 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 UId	ManagerUId	DTYPE_TEXT	100		
Middle Name	MiddleName	DTYPE_TEXT	50		
Party UId	PartyUId	DTYPE_TEXT	100		

Table 52. 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_STATUS
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_STATUS
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Related Account Integration Id	RelatedAccountIntegrationId	DTYPE_TEXT	30		
operation	operation	DTYPE_TEXT	30		System field
Household					
Household Id	HouseholdId	DTYPE_ID	30		
Household Integration Id	HouseholdIntegrationId	DTYPE_TEXT	30		
Household Name	HouseholdName	DTYPE_TEXT	50		
Household Number	HouseholdNumber	DTYPE_TEXT	100		
Household Party UId	HouseholdPartyUId	DTYPE_TEXT	100		

Table 52. Contact Relationship Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Household Status	HouseholdStatus	DTYPE_TEXT	30		Bound by HOUSEHOLD_STATUS
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
operation	operation	DTYPE_TEXT	30		System field
Organization					
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Organization	Organization	DTYPE_TEXT	50		
Organization Id	OrganizationId	DTYPE_ID	30		
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 17 describes the structure of this message.

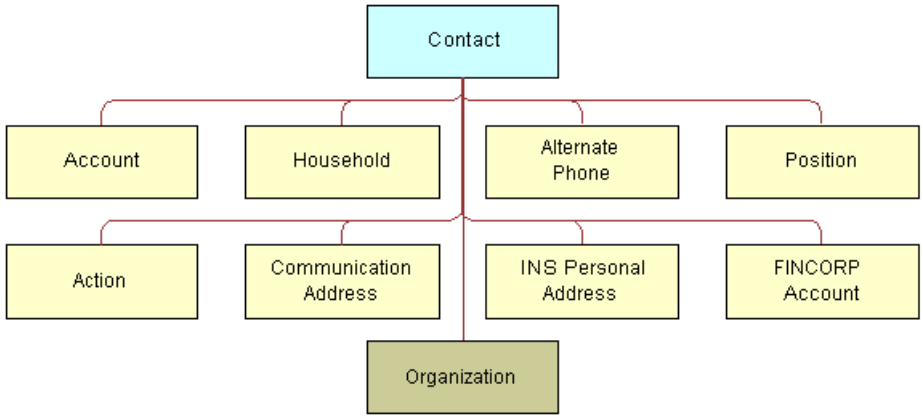

Figure 17. UCM Contact Structure

Table 53. Contact Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContact	Contact	LookUpContact	UK1:PartyUI UK2: Id
CIFContactRs	Contact	LookUpContactResponse	UK1:PartyUI UK2: Id

Table 54. Contact Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactByName	Contact	LookUpContactByName	UK1:PartyUI UK2: Id
CIFContactByNameRs	Contact	LookUpContactByNameResponse	UK1:PartyUI UK2: Id

Table 55. Contact Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactUpdate	Contact	UpdateContact	UK1:PartyUI UK2: Id
CIFContactUpdateRs	Contact	UpdateContactResponse	UK1:PartyUI UK2: Id

Table 56. Contact Upsert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactUpsert	Contact	UpsertContact	UK1:PartyUI UK2: Id
CIFContactUpsertRs	Contact	UpsertContactResponse	UK1:PartyUI UK2: Id

Table 57. Contact Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactInsert	Contact	InsertContact	UK1:PartyUI UK2: Id
CIFContactInsertRs	Contact	InsertContactResponse	UK1:PartyUI UK2: Id

Table 58. Contact Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFContactDelete	Contact	DeleteContact	UK1:PartyUI UK2: Id
CIFContactDeleteRs	Contact	DeleteContactResponse	UK1:PartyUI UK2: Id

Table 59. Contact Integration Component Fields

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

Table 59. Contact 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
Manager First Name	ManagerFirstName	DTYPE_TEXT	50		
Manager Id	ManagerId	DTYPE_ID	15		
Manager Integration Id	ManagerIntegrationId	DTYPE_TEXT	30		
Manager Last Name	ManagerLastName	DTYPE_TEXT	50		Used for picking a related entity
Manager Middle Name	ManagerMiddleName	DTYPE_TEXT	50		
Manager Primary Organization	ManagerPrimaryOrganization	DTYPE_TEXT	100		
Manager Primary Organization Id	ManagerPrimaryOrganization Id	DTYPE_ID	15		
Manager UId	ManagerUId	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 UId	PartyUId	DTYPE_TEXT	100		
Person UId	PersonUId	DTYPE_TEXT	100		
Preferred Communications	PreferredCommunications	DTYPE_TEXT	30		Bound by OFFER_MEDIA
Preferred Language Code	PreferredLanguageCode	DTYPE_TEXT	20		
Privacy Code	PrivacyCode	DTYPE_TEXT	30		Bound by PRIVACY_CODE
Social Security Number	SocialSecurityNumber	DTYPE_TEXT	11		

Table 59. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Status	Status	DTYPE_TEXT	30		Bound by CONTACT_STATUS
Suppress All Calls	SuppressAllCalls	DTYPE_TEXT	1		
Suppress All Emails	SuppressAllEmails	DTYPE_TEXT	1		
Suppress All Faxes	SuppressAllFaxes	DTYPE_TEXT	1		
Suppress All Mailings	SuppressAllMailings	DTYPE_TEXT	1		
Time Zone Id	TimeZoneId	DTYPE_ID	15		
Time Zone Name	TimeZoneName	DTYPE_TEXT	100		Used for picking a related entity.
Work Phone #	WorkPhone	DTYPE_TEXT	40		
Work Phone Extension	WorkPhoneExtension	DTYPE_TEXT	10		
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	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					

Table 59. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
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
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	HouseholdPartyUId	DTYPE_TEXT	100		

Table 59. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
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		
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		

Table 59. Contact Integration Component Fields

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

Table 59. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
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
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_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

Table 59. Contact Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
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		
Start Date	StartDate	DTYPE_DATETIME	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, InsertHoushold, and DeleteHousehold are message pairs for request and response operations. Figure 18 describes the structure of this message.

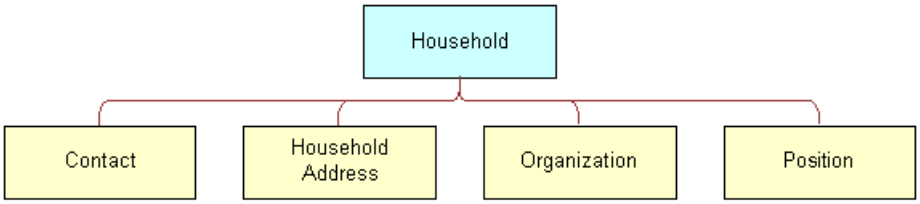

Figure 18. UCM Household Structure

Table 60. Household Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHousehold	Household	LookUpHousehold	UK1:PartyUI UK2: Id
CIFHouseholdRs	Household	LookUpHouseholdResponse	UK1:PartyUI UK2: Id

Table 61. Household Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHouseholdByName	Household	LookUpHouseholdByName	UK1:PartyUI UK2: Id
CIFHouseholdByNameRs	Household	LookUpHouseholdByNameResponse	UK1:PartyUI UK2: Id

Table 62. Household Update Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHouseholdUpdate	Household	UpdateHousehold	UK1:PartyUI UK2: Id
CIFHouseholdUpdateRs	Household	UpdateHouseholdResponse	UK1:PartyUI UK2: Id

Table 63. Household Insert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHouseholdInsert	Household	InsertHousehold	UK1:PartyUI UK2: Id
CIFHouseholdInsertRs	Household	InsertHouseholdResponse	UK1:PartyUI UK2: Id

Table 64. Houshold Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFHouseholdDelete	Household	DeleteHousehold	UK1:PartyUI UK2: Id
CIFHouseholdDeleteRs	Household	DeleteHouseholdResponse	UK1:PartyUI UK2: Id

Table 65. 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

Table 65. Household Integration Component Fields

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

Table 65. Household Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
Type	Type	DTYPE_TEXT	30		Bound by HOUSEHOLD_TYPE
operation	operation	DTYPE_TEXT	30		
Contact					
Contact Integration Id	ContactIntegrationId	DTYPE_TEXT	30		
Contact Party UId	ContactPartyUId	DTYPE_TEXT	100		
DOB	DOB	DTYPE_DATE	7		
First Name	FirstName	DTYPE_TEXT	50	✔	
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
Last Name	LastName	DTYPE_TEXT	50	✔	
M/M	MM	DTYPE_TEXT	15		Bound by MR_MS
Middle Name	MiddleName	DTYPE_TEXT	50		
Person UId	PersonUId	DTYPE_TEXT	100		
Related Contact Id	RelatedContactId	DTYPE_ID	15		
Related Date Entered Household	RelatedDateEnteredHousehold	DTYPE_UTCDATETIME	7		
Related Date Exited Household	RelatedDateExitedHousehold	DTYPE_UTCDATETIME	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					

Table 65. Household Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Note
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	✔	
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 19 describes the structure of this message.

Figure 19. UCM Personal Address Structure

Table 66. 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	Personal Address	LookUpPersonalAddress Response	1: Street Address, City,State, Postal Code

Table 67. 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 67. 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 20 describes the structure of this message.

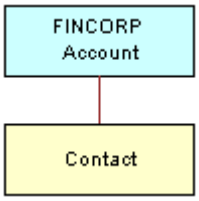

Figure 20. UCM FINCORP Account Structure

Table 68. FINCORP Account Query Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFFINCORPAccount	FINCORP Account	LookUpFINCORPAccount	UK1: Account Number
CIFFINCORPAccountRs	FINCORP Account	LookUpFINCORPAccountResponse	UK1: Account Number

Table 69. 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 69. 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 69. 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 UId	AcctGen-AuthorizedUsers-PartyUId	DTYPE_TEXT	100		
CL Customer Value	CLCustomerValue	DTYPE_TEXT	30		Bound by CL_ACCTPRFGRP_LOV
IsPrimaryMVG	IsPrimaryMVG	DTYPE_TEXT	1		
operation	operation	DTYPE_TEXT	30		System field

Account Profile

UCM Account Profile messages are used in insert, query, update, and delete operations to maintain Account Profile Information. LookUpAccount Profile and LookUptProfileByName are used for query operations. UpdateAccountProfile, InsertAccountProfile, and DeleteAccountProfile are message pairs for request and response operations. Figure 21 describes the structure of this message.

Figure 21. UCM Account Profile Structure

Table 70. Account Profile Query Integration Components

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

Table 71. Account Profile Query by Name Integration Components

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

Table 72. Account Profile Update Integration Components

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

Table 73. Account Profile Insert Integration Components

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

Table 74. Account Profile Delete Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountProfileInsert	Account	DeleteAccountProfile	UK1:PartyUId UK2: Id
CIFAccountProfileInsertRs	Account	DeleteAccountProfileResponse	UK1:PartyUId UK2: Id

Table 75. 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 75. 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 75. 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 22 describes the structure of this message.

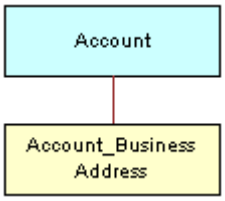

Figure 22. UCM Account Address Structure

Table 76. Account Address Query Integration Components

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

Table 77. Account Address Query by Name Integration Components

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

Table 78. Account Address Update Integration Components

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

Table 79. Account Address Insert Integration Components

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

Table 80. Account Address Delete Integration Components

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

Table 81. 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 81. 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 81. 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_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 81. 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 23 describes the structure of this message.

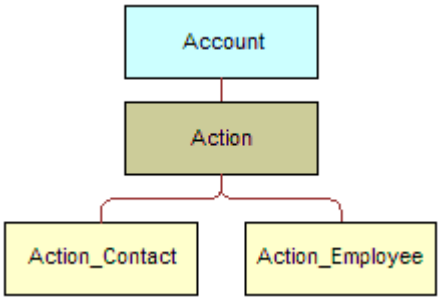

Figure 23. UCM Account Activity Structure

Table 82. Account Activity Query Integration Components

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

Table 83. Account Activity Query by Name Integration Components

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

Table 84. Account Activity Update Integration Components

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

Table 85. Account Activity Insert Integration Components

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

Table 86. Account Activity Delete Integration Components

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

Table 87. 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 87. 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 87. 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_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 87. 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 87. 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 87. 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 24 describes the structure of this message.

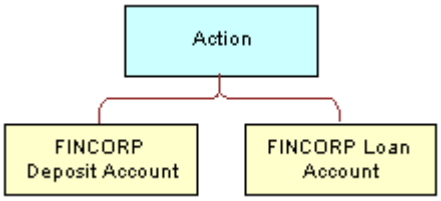

Figure 24. UCM Account Products Structure

Table 88. Account Product Query Integration Components

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

Table 89. Account Product Query by Name Integration Components

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

Table 90. Account Product Update Integration Components

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

Table 91. Account Product Insert Integration Components

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

Table 92. Account Product Delete Integration Components

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

Table 93. 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 93. 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 93. 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 93. 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 93. 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 25 describes the structure of this message.

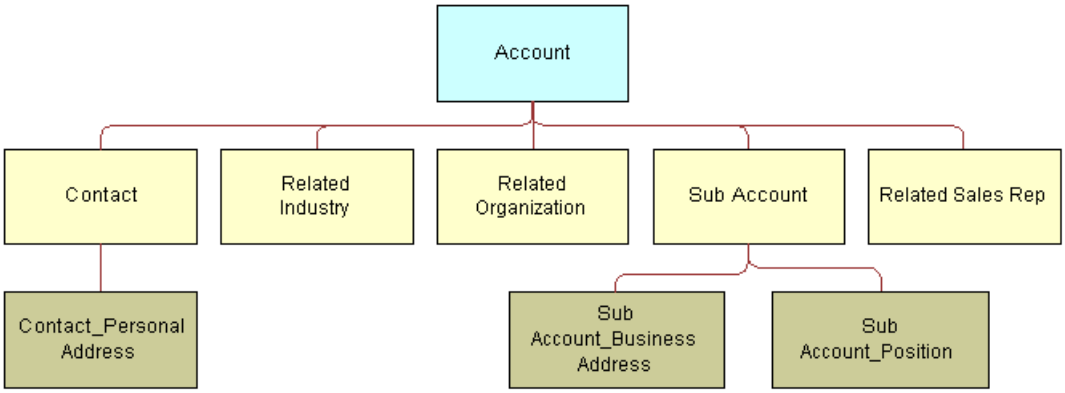

Figure 25. UCM Account Relationship Structure

Table 94. Account Relationship Query Integration Components

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

Table 95. Account Relationship Query by Name Integration Components

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

Table 96. Account Relationship Update Integration Components

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

Table 97. Account Relationship Insert Integration Components

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

Table 98. Account Relationship Delete Integration Components

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

Table 99. 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 99. 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 99. 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 99. 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	PreferredCommunication Method	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 99. 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 99. 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 99. 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 Information. LookUpAccount and LookUpByName are used for query operations. UpdateAccount, InsertAccount, UpsertAccount, and DeleteAccount are message pairs for request and response operations. Figure 26 describes the structure of this message.

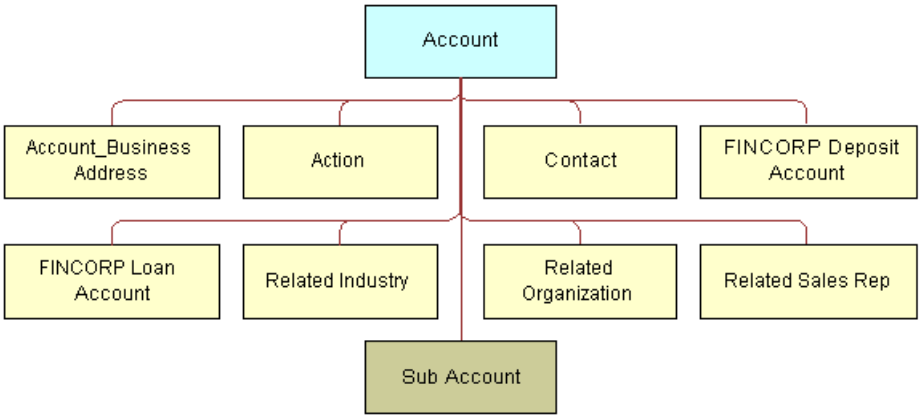

Figure 26. UCM Account Structure

Table 100. Account Query Integration Components

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

Table 101. Account Query by Name Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountByName	Account	LookUpAccountByName	UK1:PartyUId UK2: Id
CIFAccountByNameRs	Account	LookUpAccountByNameResponse	UK1:PartyUId UK2: Id

Table 102. Account Update Integration Components

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

Table 103. Account Upsert Integration Components

Integration Component	Business Component	XML Tag	User Keys
CIFAccountUpsert	Account	UpsertAccount	UK1:PartyUId UK2: Id
CIFAccountUpsertRs	Account	UpsertAccountResponse	UK1:PartyUId UK2: Id

Table 104. Account Insert Integration Components

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

Table 105. Account Delete Integration Components

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

Table 106. 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
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

Table 106. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
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	✔	
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

Table 106. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
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	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		

Table 106. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
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		
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

Table 106. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
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		
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	✔	

Table 106. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
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					
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					

Table 106. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
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
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	✔	

Table 106. Account Integration Component Fields

Name	XML Tag	Data Type	Length	Req	Notes
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		
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 27 describes the structure of these messages.

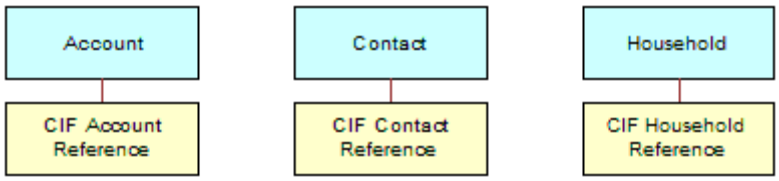

Figure 27. UCM Cross Reference Structure

Table 107. Account Cross Reference Integration Objects

Integration Component	Business Component	XML Tag	User Keys
UCMAccountExternalID	Contact	LookUpAccountExternalID	UK1:PartyUId UK2: Id
UCMAccountExternalIDRs	Contact	LookUpAccountExternalIDResponse	UK1:PartyUId UK2: Id

Table 108. Contact Cross Reference Integration Objects

Integration Component	Business Component	XML Tag	User Keys
UCMContactExternalID	Contact	LookUpContactExternalID	UK1:PartyUId UK2: Id
UCMContactExternalIDRs	Contact	LookUpContactExternalIDResponse	UK1:PartyUId UK2: Id

Table 109. Household Cross Reference Integration Objects

Integration Component	Business Component	XML Age	User Keys
UCMHouseholdExternalID	Household	LookUpHouseholdExternalID	UK1:Id UK2:IntegrationId UK3:PartyUId
UCMHouseholdExternalIDRs	Household	LookUpHouseholdExternalID Response	UK1:Id UK2:IntegrationId UK3:PartyUId

Table 110. 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			

Table 110. Account Cross Reference Integration Component Fields

Name	XML Tag	Data Type	Length	Required	Note
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		
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 111. 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 111. 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 112. 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 112. 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 113 sets forth the Business Components included in Siebel Master Data Applications.

Table 113. 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
	CUT 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 113. 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 113. 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 113. 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 113. 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 113. 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 113. 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 113. 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

B

C

K

L

M

N

O

P

W

X

Z