Siebel

Customer Hub (UCM) Master Data Management Reference

April 2026

Siebel
Customer Hub (UCM) Master Data Management Reference

April 2026

F84299-09

Copyright © 1994, 2026, Oracle and/or its affiliates.

Author: Sukanya Mishra

Contents

Get Help i

1 What's New in This Release 1

What's New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 26.3 Update 1
What's New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 25.12 Update 1
What's New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 25.7 Update 1
What's New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 25.3 Update 2
What's New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 24.4 Update 2
What's New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 23.6 Update 2
What's New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 21.11 Update 3
What's New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 21.10 Update 3
What's New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 21.9 Update 4
Whats New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 20.1 Update 4

2 Overview of Oracle Master Data Management Applications 5

Overview of Oracle Master Data Management Applications 5
Oracle Master Data Management Applications 5
About Oracle Master Data Management Applications Modules 6
Product Components Included with Oracle Master Data Management Applications 9

3 Architecture and Framework of Oracle Customer Hub (UCM) 11

Architecture and Framework of Oracle Customer Hub (UCM) 11
About Oracle Customer Hub (UCM) Framework and Interactions 11
About Oracle Customer Hub (UCM) Functions and Services 12
Administering Oracle Customer Hub 13
About Oracle Customer Hub (UCM) Architecture 13
About Oracle Customer Hub (UCM) Workflows 25
About Oracle Customer Hub (UCM) Server Components 28
About the Universally Unique Identifier 29
About Oracle Customer Hub (UCM) Concepts 29
Scenario for Deploying Oracle Customer Hub (UCM) 34

4 Configuring the List Import Function 37
Configuring the List Import Function 37
About List Management in Oracle Customer Hub (UCM) 37
Scenarios for Using the List Import Function with Oracle Customer Hub (UCM) 38
Configuring List Import Function 38
Administering the List Import Function 43

5 Configuring Source Data History and Cross-Reference Functionality 49
Configuring Source Data History and Cross-Reference Functionality 49
Configuring Source Data History and Cross-Reference Functionality for Oracle Customer Hub 49
Additions and Enhancements for Social Profile Child Object Cross Reference Functionality 53
New Applets for Source Data History Functionality 56

6 Configuring Oracle Customer Hub (UCM) to Resolve Duplicate Records 67
Configuring Oracle Customer Hub (UCM) to Resolve Duplicate Records 67
Configuring Duplicate-Resolution Functionality 67
Administering Duplication Resolution Functionality 82

7 Configuring the Survivorship Settings for Oracle Customer Hub (UCM) 87
Configuring the Survivorship Settings for Oracle Customer Hub (UCM) 87
Configuring Survivorship Settings for Oracle Customer Hub (UCM) 87
Process of Creating Survivorship Rules 97
About Default Survivorship Rules in Oracle Customer Hub (UCM) 102
Example Process of Creating a New Attribute Group for the Contacts Default Rule 102
New and Enhanced Applets for the Survivorship Engine 104

8 Configuring Publish and Subscribe Functionality 115
Configuring Publish and Subscribe Functionality 115
About Publish and Subscribe Functionality 115
Using Publish and Subscribe for Customized Integration Topologies 116
Registering Systems Connected to Oracle Customer Hub (UCM) 116
About System Publish and Subscribe Modes 118
About Outbound PROXY Services 120
Configuring System Publish and Subscribe for Oracle Customer Hub (UCM) 120
Modified Table for Oracle Customer Hub (UCM) Version 26.3 121

9 Configuring Data Decay Functionality in Oracle Customer Hub (UCM) 123
Configuring Data Decay Functionality in Oracle Customer Hub (UCM) 123
About Data Decay Functionality 123
Configuring Data Decay Functionality 124
Administering Data Decay Functionality 126

10 Configuring Oracle Customer Hub (UCM) Privacy Management 133
Configuring Oracle Customer Hub (UCM) Privacy Management 133
About Oracle Customer Hub (UCM) Privacy Management 133
Overview of Requirements for Privacy Management 135
About Oracle Customer Hub (UCM) Rules Engine Helper 135
About Privacy Management Workflow Processes 135
Process of Configuring Privacy Management 137
Administering Privacy Management 139
About HaleyAuthority in Oracle Customer Hub (UCM) Privacy Management 146
About the Business Rules Architecture in Oracle Customer Hub (UCM) Privacy Management 146
Privacy Vocabulary 146
Modules with Privacy Rules 149
Customization Methods 152
Required Customizations 153
Examples of Customizing the Sample Knowledge Base 154
About Blank Privacy Source Data and History Records 156

11 Configuring UCM Transaction Manager Callback Functions 159
Configuring UCM Transaction Manager Callback Functions 159
About UCM Transaction Manager Callback Functions 159
Scenarios for Using the UCM Transaction Manager Callback Functions 160
Enabling UCM Transaction Manager Callback Functions 162
Using UCM Utility Service Methods to Simulate External Business Service Callback Functions 166
Sample eScript for Custom Business Service After Commit 167

12 Configure Siebel UCM using Event Driven Architecture 169
Configuring Siebel UCM using the Event Driven Architecture 169
Overview of Siebel UCM using Event Driven Architecture 169
Incoming Event Flows 169
Data Steward Initiated Flows 171

Configuring Siebel UCM using Event Driven Architecture 172

13 Using Oracle Data Goverance Manager 175

Using Oracle Data Goverance Manage 175
About Oracle Data Governance Manager 175
Overview of Data Governance Manager Modules 176
Scenarios for Using Oracle Data Governance Manager 177
Configuring Cleanse Functionality 179
Viewing Mastered Records 180
Administering Batch Imports 180
Viewing Incomplete Records 182
Viewing Merge Activity 182
Viewing the Suspect Queue and Performing a Merge 182

14 Configuring the MDM Foundation and Workflows 185

Configuring the MDM Foundation and Workflows 185
Configuring Workflows and the MDM Foundation 185
Process of Configuring Inbound Communication Using Siebel EAI MQSeries Transport 199
Process of Configuring Siebel EAI JMS Transport to Integrate Data 202
Process of Configuring Siebel EAI HTTP Transport 204
About the Integration of Oracle Customer Hub (UCM) with Siebel Business Applications 205
Extending the Integration of Oracle Customer Hub (UCM) with Siebel Business Applications 206
Configuring Siebel Master Data Integration Objects 206
Cleansing Batch Data and Matching Data 212

15 Configuring Oracle Higher Education Constituent Hub 217

Configuring Oracle Higher Education Constituent Hub 217
About Oracle Higher Education Constituent Hub 217
Viewing Constituent Data in Oracle Higher Education Constituent Hub 217
Scenarios for Using Oracle Higher Education Constituent Hub 218
Configuring Oracle Higher Education Constituent Hub 219
Administering Oracle Higher Education Constituent Hub 219

16 Oracle Automotive Captive Finance Customer Hub 225

Oracle Automotive Captive Finance Customer Hub 225
About Oracle Automotive Captive Finance Customer Hub 225

Oracle Automotive Captive Finance Customer Hub Views 225

17 Oracle Case Hub 233

Oracle Case Hub 233
About Oracle Case Hub 233
Oracle Case Hub Views 235

18 Oracle Life Sciences Customer Hub 243

Oracle Life Sciences Customer Hub 243
About Oracle Life Sciences Customer Hub 243
Oracle Life Sciences Customer Hub Views 244

19 Embedded EDQ Access 255

Embedded EDQ Access 255
About Embedded EDQ Access 255
Setting Up Embedded EDQ Access 255
Using Embedded EDQ Access 255

20 Oracle Customer Hub (UCM) and Data Governance Manager Web Services Reference 257

Oracle Customer Hub (UCM) and Data Governance Manager Web Services Reference 257
Core Oracle Customer Hub (UCM) Web Services 258
OrganizationService 258
PersonService 267
GroupService 279
Financial Account Service 287
OrganizationMatchService 294
PersonMatchService 296
OrganizationCrossReferenceService 298
PersonCrossReferenceService 301
GroupCrossReferenceService 304
FinancialAssetCrossReferenceService 307
Overview of Siebel Outbound Web Services 310
Configuring Publish and Subscribe 310
Enabling or Disabling Oracle Customer Hub (UCM) Services 312
About Merge Publish Service 313

Verifying Merge Publish Settings 313
Overview of Oracle Customer Hub (UCM) Data Enrichment Web Services 314
Process of Configuring Data Enrichment 314
Third-Party Data Enrichment Fields 319
Oracle Data Governance Manager Web Services 323
SystemsRegistrationService 345
ConsolidateListImportService 346
MasterService 348
ConsolidateService 352
ShareService 358
CleanseCompletenessService 359
GovernService 366

21 Oracle Customer Hub (UCM) REST APIs Reference 373
Oracle Customer Hub (UCM) REST APIs Reference 373
About Oracle Customer Hub (UCM) REST APIs 373
Provide Access to Siebel Workflow Processes and Siebel Business Services 374
Get REST APIs Document in Open API Format 375
Override Default Values in Siebel Workflow Process Properties 375
Organization Service 375
Person Service 377
Group Service 379
Financial Asset Service 380
Merge Request Service 382
UCM REST Publish Service 383

22 Schema Changes for Oracle Customer Hub (UCM) Version 8.2 385
Schema Changes for Oracle Customer Hub (UCM) Version 8.2 385
Schema Changes for Oracle Customer Hub (UCM) for Siebel CRM 14.0 385
New Tables for Oracle Customer Hub (UCM) for Siebel CRM 14.0 385
Modified Tables for Oracle Customer Hub (UCM) for Siebel CRM 14.0 392
Schema Changes for Oracle Customer Hub (UCM) Version 8.2 392
New Tables for Oracle Customer Hub (UCM) Version 8.2 392
Modified Tables for Oracle Customer Hub (UCM) Version 8.2 406

23 Oracle Customer Hub (UCM) Data Model Tables 417
Oracle Customer Hub (UCM) Data Model Tables 417

About Oracle Customer Hub (UCM) Data Model Tables 417

24 Additional License Restrictions 423

Additional License Restrictions 423
Additional License Grants and Restrictions Regarding Oracle Master Data Management Applications 423

25 Troubleshooting Oracle Master Data Management Applications 437

Troubleshooting Oracle Master Data Management Applications 437
Troubleshooting Oracle Master Data Management Applications 437

Get Help

Preface

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

Using Oracle Applications

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

Documentation Accessibility

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

Contacting Oracle

Access to Oracle Support

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

Comments and Suggestions

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

1 What's New in This Release

What’s New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 26.3 Update

The following information lists the changes described in this version of the documentation to support this release of the software.

Topic	Description
Configuring Publish and Subscribe Functionality	New chapter. This chapter provides reference information for Oracle Customer Hub (UCM) Publish and Subscribe feature.

What’s New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 25.12 Update

The following information lists the changes described in this version of the documentation to support this release of the software.

Topic	Description
Configuring Siebel UCM using the Event Driven Architecture	Modified chapter. This chapter describes SubmitMergeRequest and Approve Scenarios.

What’s New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 25.7 Update

The following information lists the changes described in this version of the documentation to support this release of the software.

Topic	Description
Configuring Siebel UCM using the Event Driven Architecture	Modified chapter. This chapter describes about event driven architecture support for Kafka partitions and Query in Siebel UCM.

What’s New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 25.3 Update

The following information lists the changes described in this version of the documentation to support this release of the software.

Topic	Description
Oracle Customer Hub (UCM) REST APIs Reference	New topic. This chapter explains scalable and flexible integration approach using REST APIs for Inbound UCM processes using JSON format and outbound REST to publish JSON format messages to external systems.

What’s New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 24.4 Update

The following information lists the changes described in this version of the documentation to support this release of the software.

Topic	Description
Configuring Siebel UCM using the Event Driven Architecture	New topic. This chapter describes how to configure Siebel UCM using Event Driven Architecture (Siebel CRM Event Pub Sub framework).

What’s New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 23.6 Update

The following information lists the changes described in this version of the documentation to support this release of the software.

Topic	Description
Configuring UCM Transaction Manager Callback Functions	New topic. Oracle Customer Hub (UCM) utilizes UCM Transaction Manager to commit the records as Siebel Database Transaction. This chapter describes how to configure UCM Transaction Manager’s callback functions for Oracle Master Data Management Applications.

What's New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 21.11 Update

The following topic has been added to improve the technical accuracy of this guide:

- *Modifying Process Properties for UCM Batch Process Workflow*

The following topic has been revised to improve the technical accuracy of this guide:

- *Setting List Import System Preferences*

A number of topics have been deleted because the information they contained is obsolete.

The following topics have been revised to remove references to obsolete information:

- *Siebel Industry EAI Architecture*
- *Siebel Enterprise Integration Manager and Integration of Oracle Customer Hub (UCM)*
- *UCM Async Realtime Publish Workflow*
- *UCM Batch Process*
- *Oracle Customer Hub (UCM) Data Steward Functions*
- *Setting List Import System Preferences*
- *Real-Time Mode*
- *Setting System Privileges for Oracle Master Data Management Applications*
- *Configuring Named Subsystems*
- *Configuring Named Subsystems*
- *About the Integration of Oracle Customer Hub (UCM) with Siebel Business Applications*
- *UCM Integration Objects*
- *Publish Subscribe Service*
- *Troubleshooting Oracle Master Data Management Applications*

What's New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 21.10 Update

No new features have been added to this guide for this release. The following topics were added to improve the technical accuracy of this guide:

- *Configuring Oracle UCM to Send User Keys for Child Records to xternal Rules Engine*
- *About the Exact Match Functionality*

The following topic was revised to improve the technical accuracy of this guide:

- *Configuring Exact Match Functionality in Oracle Customer Hub (UCM)*

What's New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 21.9 Update

The following information lists the changes described in this version of the documentation to support this release of the software.

Topic	Description
Configuring Survivorship to Use An External Rules Engine	Modified topic. This topic has been renamed from Configuring Survivorship to Use Haley Rules Engine to Configuring Survivorship to Use an External Rules Engine because Haley Rules Engine is obsolete. The text in the topic has also been updated to remove references to Haley Rules Engine.
Example of Using Survivorship Rules	Modified topic. The text has been updated to remove a reference to Haley Rules Engine.
Configuring Survivorship to Use An External Rules Engine *Organization Match Service* *PersonMatchService*	Modified topics. Search Software America (SSA) is no longer supported as a data quality vendor. References to SSA in these topics have accordingly been replaced with references to Oracle Enterprise Data Quality (EDQ).

Additional Changes

The following documentation changes affect the organization of this guide:

- *Configuring the Survivorship Settings for Oracle Customer Hub (UCM)*. The following topics have been removed from this chapter because Haley Rules Engine is obsolete:
  - Example of Configuring Survivorship for Haley Rules Engine
  - Example of Configuring Any Child Entity Survivorship for Haley Rules Engine
- *Configuring Survivorship Custom Date Fields*. The procedure to activate the advanced survivorship knowledge base has been removed from this topic because the procedure is obsolete.

Whats New in Oracle Customer Hub (UCM) Master Data Management Reference, Siebel CRM 20.1 Update

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

2 Overview of Oracle Master Data Management Applications

Overview of Oracle Master Data Management Applications

This chapter provides a brief overview of Oracle Master Data Management Applications, including primary file descriptions, modules, and product components. It includes the following topics:

- *Oracle Master Data Management Applications*
- *About Oracle Master Data Management Applications Modules*
- *Product Components Included with Oracle Master Data Management Applications*

Oracle Master Data Management Applications

Oracle Master Data Management Applications provide a platform that creates and maintains the primary record for an organization's enterprise-wide customer information. They consist of the following primary file modules:

- Oracle Customer Hub (UCM)
- Oracle Activity Hub
- Oracle Marketing Hub
- Oracle Sales Hub
- Oracle Service Hub
- Oracle Field Service Hub
- Oracle Privacy Management Policy Hub
- Oracle Higher Education Constituent Hub

Oracle Customer Hub (UCM) is the base module on which all other primary file modules build. These modules add extended enterprise-wide information capabilities. Oracle Customer Hub (UCM) module consists of a set of Siebel CRM data-model tables that store customer data. The other modules extend this concept to hold additional marketing, sales, and service data. They are part of Oracle's Master Data Management family of applications. For information about Oracle Master Data Management Applications, see *About Oracle Master Data Management Applications Modules*.

When Oracle Master Data Management Applications are used to maintain primary records of an organization's data, one or more of the modules interact with back-office systems and Oracle's Siebel Business Applications deployments. They provide a unified data set for an organization's multiple channels, lines of business, and applications.

Oracle Customer Hub Add-on B2B

Oracle Customer Hub Add-on B2B for Siebel CRM and Oracle E-Business Suite is a bundle that includes two components:

- **Oracle Customer Hub B2B.** If you are licensed for Oracle Customer Hub B2B, then your use of the bundled programs is defined in your license order. Your use is limited to the component that you are licensed for, and you do not have rights to use Oracle Customer Data Hub.
- **Oracle Customer Data Hub.** If you are licensed for Oracle Customer Data Hub, then your use is limited to the component that you are licensed for, and you do not have rights to use Oracle Customer Hub B2B.

Oracle Customer Hub B2C

Oracle Customer Hub Business-to-Consumer (B2C) is a bundle that includes two components:

- **Oracle Customer Hub B2C.** If you are licensed for Oracle Customer Hub B2C, then your use of the bundled programs is defined in your license order. Your use is limited to the component that you are licensed for, and you do not have rights to use Oracle Customer Data Hub.
- **Oracle Customer Data Hub.** If you are licensed for Oracle Customer Data Hub, then your use is limited to the component that you are licensed for, and you do not have rights to use Oracle Customer Hub B2C.

Oracle Customer Hub Data Steward

Oracle Customer Hub Data Steward is a product that includes two components:

- **Oracle Customer Data Steward.** If you are licensed for Oracle Customer Hub Data Steward, each user has a right to use Oracle Customer Hub Data Steward and Oracle Customer Data Librarian. Your use is limited to the component that you are licensed for, and you do not have rights to use Oracle Customer Data Librarian.
- **Oracle Customer Data Librarian.** If you are licensed for Oracle Customer Data Librarian, then your use is limited to the component that you are licensed for, and you do not have rights to use Oracle Customer Hub Data Steward.

About Oracle Master Data Management Applications Modules

Oracle Master Data Management Applications include the base module, Oracle Customer Hub (UCM), and the following additional modules comprising various subsets of enterprise-wide customer related data:

- *Oracle Customer Hub (UCM)*
- *Oracle Activity Hub*
- *Oracle Marketing Hub*
- *Oracle Sales Hub*
- *Oracle Service Hub*

- *Oracle Field Service Hub*
- *Oracle Automotive Captive Finance Customer Hub*
- *Oracle Case Hub*
- *Oracle Life Sciences Customer Hub*
- *Privacy Management Policy Hub*

Oracle Customer Hub (UCM)

Oracle Customer Hub (UCM) is configured to store a clean and unified profile for enterprise customers, partners, and prospects. Traditional customer data, such as accounts, contacts, households, partners, prospects, and agent data is included as well as customer relationship information, address information, and asset information. Oracle Customer Hub (UCM) 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.

Oracle Customer Hub (UCM) 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.

Oracle Activity Hub

Oracle Activity Hub module is an application extension to Oracle Customer Hub (UCM) 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, assets, notes, agreements, entitlements, billing accounts, demographic information, business profiles, customer preferences, credit profiles, activities history, and analytical and segmentation information.

Oracle Marketing Hub

Oracle Marketing Hub module is an application extension to Oracle Customer Hub (UCM) 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.

Oracle Sales Hub

Oracle Sales Hub module is an application extension to Oracle Customer Hub (UCM) 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.

Oracle Service Hub

Oracle Service Hub module is an application extension to Oracle Customer Hub (UCM) 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.

Oracle Field Service Hub

Oracle Field Service Hub module is an application extension to Oracle Customer Hub (UCM) 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.

Oracle Automotive Captive Finance Customer Hub

Oracle Automotive Captive Finance Customer Hub is a bundle of Oracle Customer Hub (UCM) products that together provide a customer hub solution for the automotive captive finance industry. For more information, see *Oracle Automotive Captive Finance Customer Hub*.

Oracle Case Hub

Oracle Case Hub is a bundle of Oracle Customer Hub (UCM) products that together provide a customer-hub solution where case management is the primary focus of the solution. For more information, see *Oracle Case Hub*.

Oracle Life Sciences Customer Hub

Oracle Life Sciences Customer Hub is a bundle of Oracle Customer Hub (UCM) products that together provide a customer-hub solution for the life sciences industry. For more information, see *Oracle Life Sciences Customer Hub*.

Privacy Management Policy Hub

The Privacy Management Policy Hub module is an application extension to Oracle Customer Hub (UCM) that creates a central storage location for privacy rules. For more information, see *Configuring Oracle Customer Hub (UCM) Privacy Management*.

Oracle Higher Education Constituent Hub

Oracle Higher Education Constituent Hub module provides for the needs of higher-education institutions, by maintaining a single source record of biographical and demographic data collected from many different legacy systems. For more information, see *Configuring Oracle Higher Education Constituent Hub*.

Product Components Included with Oracle Master Data Management Applications

Certain product components of Siebel Business Applications are included with Oracle Master Data Management Applications. They provide or add functionality to the primary files. Only the Siebel Business Applications required to run Oracle Master Data Management Applications are included with the Oracle Master Data Management Applications base product. The following Siebel Business Applications are included with Oracle Master Data Management Applications:

- **Siebel Application Object Manager.** Siebel Application Object Manager hosts the Business Objects layer and the Data Objects layer of Siebel Business Application. See *Siebel System Administration Guide* for more 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 more information.
- **Siebel Enterprise Integration Manager (Siebel EIM).** Siebel Enterprise Integration Manager manages the exchange of data between Siebel database tables and other back-office databases. Siebel Enterprise Integration Manager is used for bulk data imports, exports, merges, and deletions. See *Siebel Enterprise Integration Manager Administration Guide* for more information.
- **Siebel Business Process Designer.** Siebel Business Process Designer is a customizable business application that enables you to design, manage, and enforce your business processes. It enables you to design complex workflow processes and automate the enforcement of business policies and procedures. See *Siebel Business Process Framework: Workflow Guide* for more information.
- **Siebel Entity Relationship Designer.** Siebel Entity Relationship Designer is a visual design tool that enables 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 more information.

Based on the software that you purchased, one or more of the following Siebel Business Applications might also be relevant to the implementation of Oracle Master Data Management Applications:

- **Siebel Replication Manager.** Siebel Replication Manager distributes full schema and data replicas to support subordinate deployments of Siebel Business Applications. See *Siebel Remote and Replication Manager Administration Guide* for more information.
- **Siebel Business Rules Developer (HaleyAuthority).** Siebel Business Rules Developer provides a platform for users to develop new or customize existing business rules for company-specific privacy management requirements. It is integrated into the Oracle Customer Hub (UCM) Privacy Management module. For more information, see *Siebel Business Rules Administration Guide* , version 8.1.

- **Siebel Assignment Manager.** Siebel Assignment Manager distributes and assigns entities, such as opportunities or service requests to individuals, positions, or organizations. The entities are based on defined assignment rules. See *Siebel Assignment Manager Administration Guide* for more information.
- **Siebel Tools.** Siebel Tools is an integrated development environment for configuring and customizing various aspects of Siebel Business Applications, including Data Layer, Object Layer, User Interface Layer, and publish and subscribe services. See *Configuring Siebel Business Applications* for more information.
- **Siebel Data Quality.** Siebel Data Quality is a separately licensed item that assists enterprises by standardizing their contact, account, and prospect data by data matching and data cleansing. See 1515881.1 (Doc ID) *Data Quality Guide for Oracle Customer Hub* on My Oracle Support for more information.
- **Siebel Audit Trail.** Siebel Audit Trail creates a history of the changes that have been made to various types of information in Siebel Business Applications. See *Siebel Applications Administration Guide* for more information.

3 Architecture and Framework of Oracle Customer Hub (UCM)

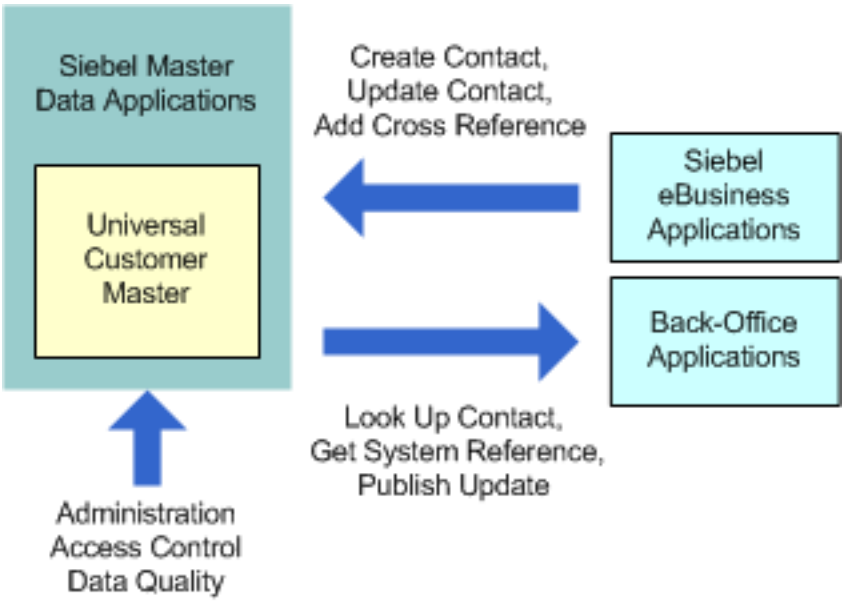
Architecture and Framework of Oracle Customer Hub (UCM)

This chapter describes a scenario for a financial institution. It also provides background information on the framework, architecture, integration, and interactions of Oracle Customer Hub. It includes the following topics:

- *About Oracle Customer Hub (UCM) Framework and Interactions*
- *About Oracle Customer Hub (UCM) Functions and Services*
- *Administering Oracle Customer Hub*
- *About Oracle Customer Hub (UCM) Architecture*
- *About Oracle Customer Hub (UCM) Workflows*
- *About Oracle Customer Hub (UCM) Server Components*
- *About the Universally Unique Identifier*
- *About Oracle Customer Hub (UCM) Concepts*
- *Scenario for Deploying Oracle Customer Hub (UCM)*

About Oracle Customer Hub (UCM) Framework and Interactions

Oracle Customer Hub (UCM) forms the primary application and database of an organization's data. It interacts with back-office systems and deployments of Siebel Business Applications to provide different organizational business units with consistent and timely data. A variety of functionality is available between Oracle Customer Hub (UCM) Applications and Siebel Business Applications or back-office applications using enterprise integration and the regular links to Siebel Business Applications. This relationship is shown in the following diagram.



For information about the functionality included with Oracle Customer Hub (UCM), see the following:

- For information about functions and services, see *About Oracle Customer Hub (UCM) Functions and Services*.
- For information about the administrative functionality of Oracle Customer Hub (UCM), see *Administering Oracle Customer Hub*.
- For information about the technical integration aspects used with Oracle Customer Hub, see *Oracle Customer Hub (UCM) Prebuilt Business Services*.

About Oracle Customer Hub (UCM) Functions and Services

The database tables and Siebel Connector for Master Data Applications support insert, delete, update, and query functionality on primary data, such as customers, products, and so on. 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 Oracle Customer Hub. For more information on Siebel Connector for Master Data Applications, see *About Oracle Customer Hub (UCM) Architecture* and *Oracle Customer Hub (UCM) Prebuilt Business Services*.

Oracle Customer Hub (UCM) also provides the following services through configuration:

- **Publish.** Publish data records from Oracle Customer Hub (UCM) to subscribing applications. For more information on administering this functionality, see *Setting System Privileges for Oracle Master Data Management Applications*.
- **Request and reply.** Other applications send a request for data from Oracle Customer Hub. For more information on configuring request and reply, see *Setting System Privileges for Oracle Master Data Management Applications* **.**

Services can be configured to operate in real-time mode or batch mode. Oracle Customer Hub (UCM) uses a variety of integration techniques to allow the operation of functions and services between other applications and themselves. See *About Oracle Customer Hub (UCM) Architecture* for more information.

Administering Oracle Customer Hub

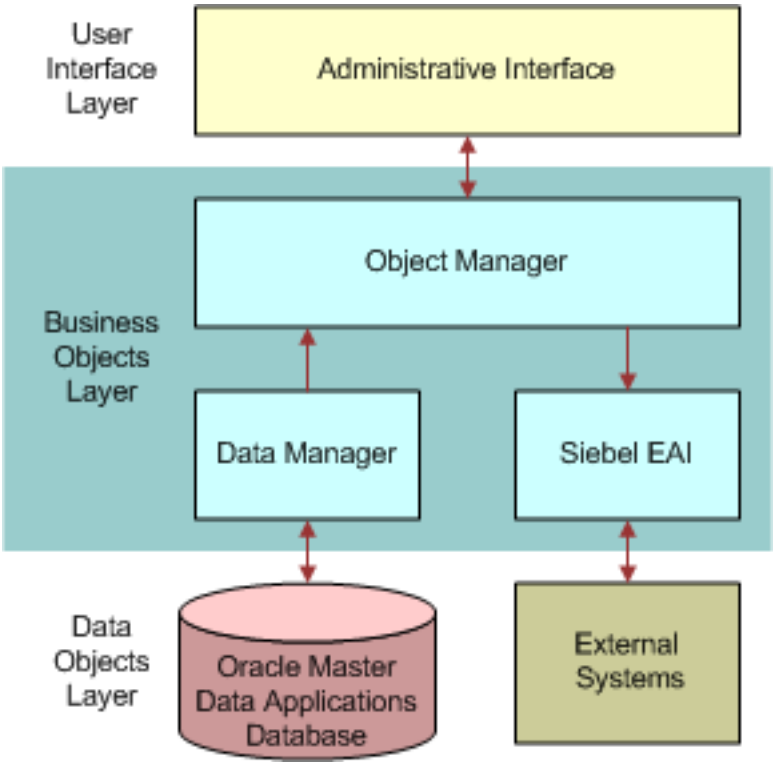
Oracle Customer Hub (UCM) is administered through Oracle Customer Hub Data Steward user interface screens. However, only those screens necessary for viewing and administering the information contained within Oracle Customer Hub (UCM) are included. Only administrators have access to these screens. The administration screens are used to make manual changes to the data, make decisions on suspect information, and set up rules and policies for managing data in Oracle Customer Hub (UCM).

Access control is enforced at the user-interface level for Oracle Customer Hub (UCM) administration screens. Oracle Customer Hub (UCM) messages have full visibility of the data; that is, no access control mechanism is applied to the messages, but they are subject to system privilege checks when accessing Oracle Customer Hub. For more information on configuring system privileges, see *Setting System Privileges for Oracle Master Data Management Applications*. For information on access control, see *Configuring Siebel Business Applications* and *Siebel Security Guide* .

About Oracle Customer Hub (UCM) Architecture

Oracle Customer Hub (UCM) is based on the n-tier object architecture of Siebel Business Applications in which the user interface, business logic, and data are separated and layered. Each tier contains a set of objects and components, which allows a high degree of reuse. The consolidated object layer enables uniform communication between channels and interfaces. This architecture also allows flexible deployments and integration with new business processes and systems. The following figure shows the Oracle Customer Hub (UCM) architecture, which has the following layers:

- **User Interface Layer.** Containing the Administrative Interface.
- **Business Objects Layer.** Containing the Object Manager, Data Manager, and Siebel EAI.
- **Data Objects Layer.** Containing the Oracle Master Data Applications Database and External Systems.



Oracle Customer Hub (UCM) architecture is designed for extreme fault tolerance with zero downtime. The database platform includes the following:

- 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 operating 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.

Oracle Customer Hub (UCM) User Interface Layer

The user interface layer for Oracle Customer Hub (UCM) Applications is interactive and customizable. It consists of two parts:

- **Physical user interface layer.** It includes templates and tags that render the user interface,
- **Logical user interface objects layer.** It 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 does not contain any business logic. For more information about the user interface layer, see *Configuring Siebel Business Applications* . The user interface layer is used for the administration of the Siebel Business Applications. For more information, see *Administering Oracle Customer Hub*.

Oracle Customer Hub (UCM) Business Objects Layer

The business logic layer for Oracle Customer Hub (UCM) contains object abstractions of entities and represents internal and external data. The logic layer also allows for complex entity relationships. For more 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 an Oracle Customer Hub (UCM) Integration instance.

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 more information about the business objects and data objects layer, see *Configuring Siebel Business Applications* .

Siebel Industry EAI Architecture

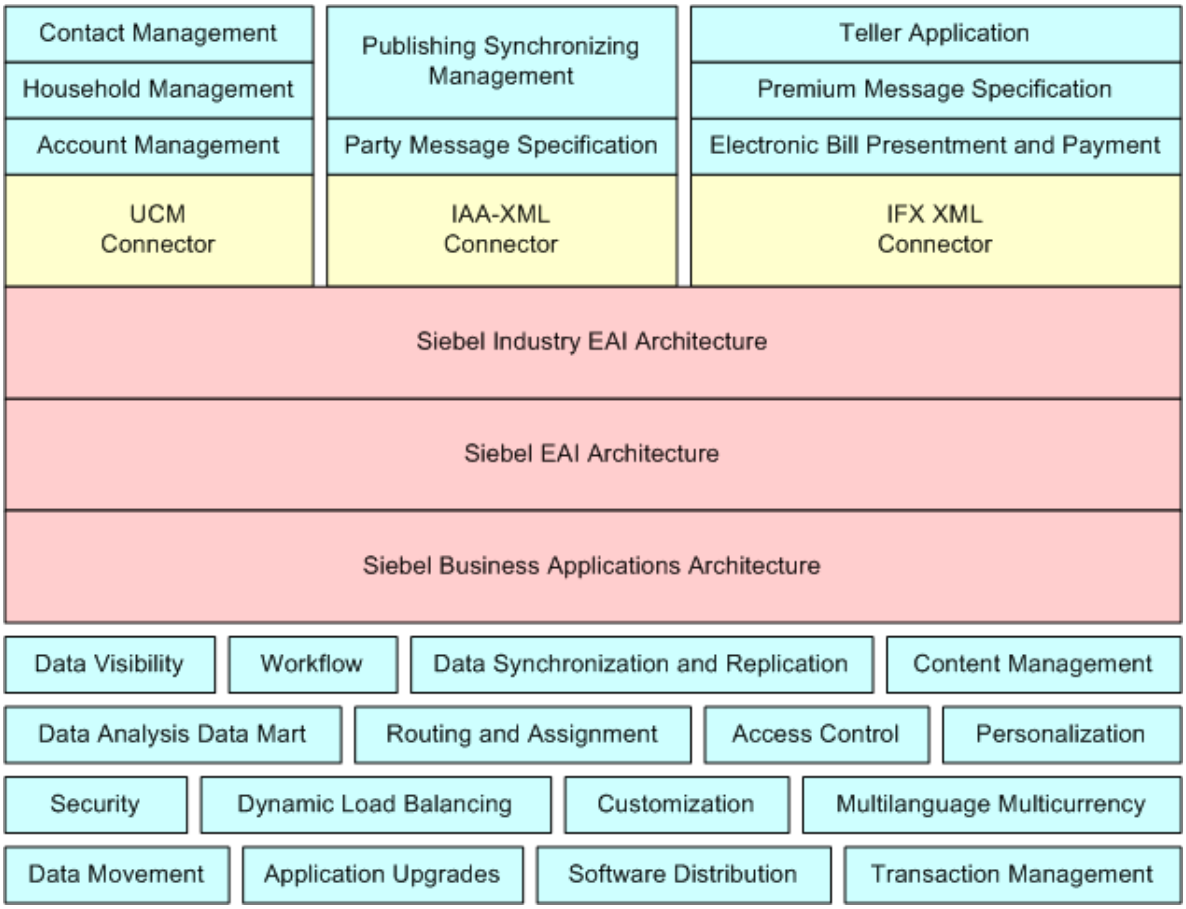
Siebel Industry EAI architecture is built to work with the Siebel Enterprise Application Integration (EAI) architecture and to support XML-based messaging communication infrastructure. Different applications require integration using messaging mechanisms. Connectors must be built to support various industry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

The connectors are based on the Siebel Industry EAI framework.

The following figure illustrates the high-level architecture of the Siebel Industry EAI and the standard connectors.



Siebel Connector for Master Data Applications is based on Siebel XML standards in addition to a predefined Oracle Customer Hub (UCM) 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. Siebel Connector for Master Data Applications provides functions such as:

- Handling message headers
- 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-time mode and batch mode

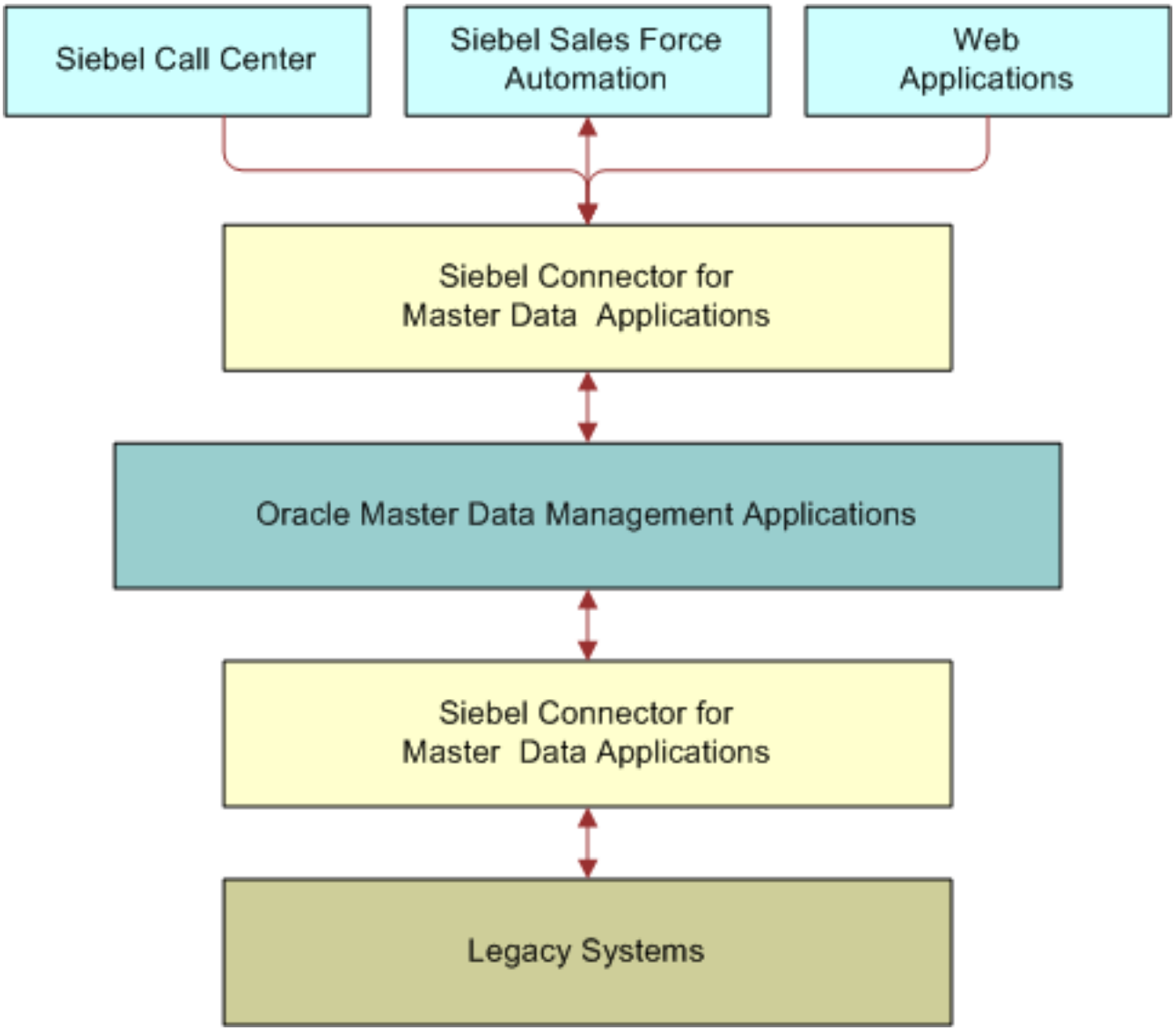
Siebel Connector for Master Data Applications includes several business services. For more information on these services, see *Oracle Customer Hub (UCM) Prebuilt Business Services*.

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. The following figure illustrates inbound (Receive-Send) business process flow. Some of the business process flows might constitute messages published by Oracle Customer Hub (UCM), such as <createPerson>, <createOrganization>, <updatePerson>, <updateGroup>, <deleteOrganization>, <getOrganization>, <getFinancialAsset>, and so on. These messages are included in Siebel Master Data Applications for Industry Applications.

The business process 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 a subscribing application.

The business process flow is displayed in the following figure.



Inbound Data Flow

An inbound business process flow starts with a receiver server component, such as the MQSeries or HTTP Receiver. The receiver server component runs in the background continuously waiting for messages to arrive from external applications. After receiving a Simple Object Access Protocol (SOAP) message, the receiver then calls 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 reads 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 in the hierarchy nodes and sends it to the UCM Security Manager.

   > **Note:** The Dispatcher Map is shipped as part of Oracle Customer Hub (UCM). For information, see *UCM Dispatcher Map Integration Objects*.

   The UCM Security Manager takes the XML instance, verifies subscribing system for registration with Oracle Customer Hub, checks the operation identified in each body section for the privileges. It then attaches any failed privilege bodies to the FailSecurityHierarchy before sending it to the UCM Converter.
3. UCM Converter takes the XML instance and processes individual sections of the instance while converting each subtree into external integration object instances before sending it to UCM Transaction Manager.
4. 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.
   - If the EnableRealtimeInsert user property is set to False, then insert requests are queued in the Source Data History table.
   - Update operations invoke Oracle Customer Hub (UCM) Survivorship Engine or Oracle Customer Hub (UCM) Data Quality Manager, if enabled.
5. If UCM Publish and 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 movement and transformation of data to and from Siebel Business Applications. You create the workflow processes, using a graphical user interface provided with Siebel Business Applications, called Siebel Business Process Designer. For information on workflows and Siebel Business Process Designer, see *Siebel Business Process Framework: Workflow 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 Oracle Customer Hub (UCM) Integration. Supporting integration objects for Oracle Customer Hub (UCM) business processes are shipped with Oracle Customer Hub. 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 connecting to the database, connecting 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 connect with external systems within an organization's application

network. For more information on business services, *Integration Platform Technologies: Siebel Enterprise Application Integration* .

Oracle Customer Hub (UCM) Data Objects Layer

Oracle Customer Hub (UCM) Applications support leading databases on various platforms. These platforms include the following: IBM DB2 for Windows and UNIX, DB2 for OS/390 and z/OS. It also includes Oracle Database and Microsoft SQL Server). The data layer can scale vertically (with the size of the server) and horizontally with database parallelization. For more information about the data objects layer, see *Configuring Siebel Business Applications* .

Oracle Customer Hub (UCM) Prebuilt Business Services

Oracle Customer Hub (UCM) includes the following prebuilt business services:

- *UCM Account Source Data and History Service*
- *UCM Batch Manager*
- *UCM Contact Source Data and History Service*
- *UCM Converter*
- *UCM Cross-Reference Service*
- *UCM Data Decay Service*
- *UCM Data Governance Manager*
- *UCM Data Governance Manager Service*
- *UCM Data Mgmt Util Service*
- *UCM Data Quality Manager*
- *UCM Data Steward UI Service*
- *UCM Dispatcher*
- *UCM FINCORP Account Source Data and History Service*
- *UCM FINCORP Account Source Data and History Service*
- *UCM Household Source Data and History Service*
- *UCM List Import Services*
- *UCM File Manager Service*
- *UCM Merge Service*
- *UCM Publish/Subscribe Service*
- *UCM Security Manager*
- *UCM Source Data and History Service*
- *UCM Survivorship Engine*
- *UCM System Task Service*
- *UCM Transaction Manager*

- *UCM UI Interaction Service*
- *UCM Unmerge Service*
- *UCM Utility Service*
- *Transport Adapter*

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 Oracle Customer Hub (UCM) Architecture*.

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 Batch Manager

UCM Batch Manager processes insert requests queued in the Source Data History (SDH) table in batch mode. It also processes data in batch mode during the initial loading through Siebel Enterprise Integration Manager.

You can enable UCM Batch Manager to allow multiple view modes, such as viewing records across multiple organizations, by setting the ViewMode User Property in the UCM Batch Manager. For configuration instructions, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

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 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 Cross-Reference Service

The UCM Cross-Reference Service handles different operations on the UCM Cross-Reference records.

UCM Data Decay Service

The UCM Data Decay Service handles data decay related operations. It resets the decay metrics when a field is updated. It prepares inputs to invoke the rules engine and parses the rule outputs, and it prepares the inputs to notify the data steward about decayed data.

UCM Data Governance Manager

This service provides an interface to support the handling of Create, Retrieve, Update and Delete (CRUD) operations on the cross-reference information for the primary business objects (Account, Contact).

UCM Data Governance Manager Service

UCM Data Governance Manager Service is a business service that provides an interface and also the implementation for the handling of the requests from Oracle Data Governance Manager which is based on Oracle Application Development Framework. The methods of this business service address the categories of Master, Consolidate, Share, Cleanse and

Govern areas. These methods have been registered in the Siebel Web Service framework as Inbound Web service operations for the following Web services:

- MasterService
- ConsolidateService
- ConsolidateListImportService
- ShareService
- GovernService
- CleanseCompletenessService
- SystemsRegistrationService

For more information on these Web services, see *Oracle Customer Hub (UCM) and Data Governance Manager Web Services Reference*

UCM Data Mgmt Util Service

The UCM Data Mgmt Util Service is invoked for Guided Merge tasks. It validates customer inputs from Task Based UI applets and prepares inputs for UCM Merge service.

UCM Data Quality Manager

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

UCM Data Quality Business Services

UCM Data Quality includes the following business services:

- DataEnrichment
- ISS System Services
- ISS Utility Services
- TestDqSync

UCM Data Steward UI Service

The UCM Data Steward UI Service handles UI operations that the data steward performs, including preparing for Merge and Unmerge requests and removing the Pending status from suspect records.

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 does the following actions:

- Parses the envelope of the message
- Converts the envelope to the header property set
- 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 FINCORP Account Source Data and History Service

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

UCM Household Source Data and History Service

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

UCM List Import Services

Provides the main functions for the List Import service to process import jobs, validate data, and handle errors. This business service also includes functions to process List Import Web service requests.

UCM File Manager Service

Use in List Import to process import files and write records into Oracle Customer Hub (UCM) staging tables (SDH tables).

UCM Merge Service

The UCM Merge Service handles all the UCM Merge related operations. It validates merge requests, submits valid merge request for further merge processing, prepares a History record to be stored in the Source Data and History tables, calls survivorship to decide on the winning fields, prepares inputs for merging child objects, and purges merge requests on a user specified status.

UCM Publish/Subscribe Service

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

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. The output is based on the Business Object type that is 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 Oracle Customer Hub (UCM) message specification and sends them to the registered system through registered protocol types.

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

The UCM Publish/Subscribe Service has the following default user properties:

- EnableCrossReference[FALSE]
- ReloadSystemsCounter[10]
- WorkflowName[UCM Async Realtime Publish Workflow]

You can enable UCM Publish/Subscribe Service to allow multiple view modes, such as viewing records across multiple organizations, by setting the ViewMode User Property in the UCM Publish/Subscribe Service. For configuration instructions, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

UCM Security Manager

The UCM Security Manager acts as the gatekeeper of Oracle Customer Hub (UCM) Application. It verifies the System ID in the message header against the System Registrations in Oracle Customer Hub (UCM) before it can perform the 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 Oracle Customer Hub (UCM) for the particular object and operation. Any fail operation instances in the body are removed from the XMLHierarchy and attached to the FailSecurityHierarchy.

UCM Source Data and History Service

The UCM Source Data and History Service is responsible for most of the operations performed against the UCM Source Data and History tables, including saving the source record, deleting the history record, and so on.

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 Oracle Customer Hub (UCM) Survivorship Rules*.

UCM System Task Service

Used in List Import to create the import job status or update the import job status or both.

UCM Transaction Manager

The UCM Transaction Manager executes operations specified in Oracle Customer Hub (UCM) 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 Oracle Customer Hub (UCM) Application command elements to Siebel Adapter actions and combines return results as a single property set.

You can add additional business logic or processing to take place in the UCM Transaction Manager after an inbound XML message is received and processed, but before it is committed.

You can enable UCM Transaction Manager to allow multiple view modes, such as viewing records across multiple organizations, by setting the ViewMode User Property in the UCM Transaction Manager. For configuration instructions, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

UCM UI Interaction Service

UCM UI Interaction Service creates a Universally Unique Identifier (UUID) during the insert process. It 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.

> **Note:** When this service is enabled, UCM survivorship rules are not initiated, and data entered by users is not corrected. Users are responsible for accurate data entry.

UCM Unmerge Service

The UCM Unmerge Service is the interface that calls the UCM Unmerge functionality.

UCM Utility Service

The UCM Utility Service contains the utility functions that Oracle Customer Hub (UCM) operations require, including Integration Object to Business Component Hierarchy conversion utilities, constructing Hierarchy property set based on the row ID, and a method to retrieve an Oracle Customer Hub (UCM) Customer Data Management (CDM) parameter, and so on.

Transport Adapter

Transport Adapter is a prebuilt business service providing an interface between Siebel Business Applications and external applications. Transports allow Oracle Customer Hub (UCM) 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 information on transport adapters, see *Transports and Interfaces: Siebel Enterprise Application Integration* .

Siebel Enterprise Integration Manager and Integration of Oracle Customer Hub (UCM)

Siebel Enterprise Integration Manager is an important component of Oracle Customer Hub. It performs bulk imports and exports data residing in other back-office systems into Oracle Customer Hub (UCM) and consolidates and replicates information from these sources. Unlike Siebel Enterprise Application Integration, Siebel Enterprise Integration Manager does not go through an application object manager, but the data moves from the EIM_UCM_ORG interface table directly into the SDH (S_UCM_ORG) tables. Siebel Enterprise Integration Manager also provides a schema abstraction and has high throughput. Some features of Siebel Enterprise Integration Manager include:

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

For batch integration with external applications, Oracle Customer Hub (UCM) provides the following Customer Hub-specific EIM tables to facilitate the loading of selected primary objects into Oracle Customer Hub's Source Data and History tables. These records can then be processed by Oracle Customer Hub (UCM) data management tasks later:

- EIM_PTYUCM_DTL
- EIM_UCM_ADRPER
- EIM_UCM_ASSET
- EIM_UCM_ASTCON
- EIM_UCM_CON
- EIM_UCM_CONCHLD
- EIM_UCM_OGPCHD
- EIM_UCM_ORG
- EIM_UCM_ORGCHLD
- EIM_UCM_ORGGRP
- EIM_UCM_PRIVCY

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

Generation of Party_UID

Siebel Enterprise Integration Manager and Oracle Customer Hub (UCM) 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 (Organization, Person, Financial Asset, or Group) Customer Profile Integration SOAP Process.** In this scenario, the user must populate the <ID> integration object component field value in the SOAP 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 SOAP message. The UCM (Organization, Person, Financial Asset, or Group) Customer Profile Integration SOAP Process then generates the PARTY_UID based on UCM_EXT_ID.
- **Loading data to UCM SDH table using Siebel Enterprise Integration Manager.** 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 Process workflow then generates the PARTY_UID based on the UCM_EXT_ID.

For more information on the PARTY_UID, see *Siebel Enterprise Integration Manager Administration Guide* .

About Oracle Customer Hub (UCM) Workflows

Workflows are a feature of Oracle Customer Hub (UCM). 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 Framework: Workflow Guide* .

Use the Workflow Process Designer view in the Administration - Business Process screen to view the diagrams of the following workflows. The following workflows describe the business processes for Oracle Customer Hub (UCM):

- *UCM Account Data Decay Process*
- *UCM Async Realtime Publish Workflow*
- *UCM Batch Process*
- *UCM Contact Data Decay Process*
- *UCM Daily Publish*
- *UCM Import File Processing*
- *UCM Master List Import Process*
- *UCM Process Merge Request*

UCM Account Data Decay Process

This workflow calls HaleyAuthority decay business rules to evaluate an account record's decay scores and raises an alarm when the decay score falls below a preset threshold. It runs in batch mode using the Workflow Process Batch Manager server component. You can use search specifications to limit the number of records that are evaluated.

UCM Async Realtime Publish Workflow

This workflow is invoked as part of a server request to perform asynchronous real-time publishing. It is called from either the UCM (Organization, Person, Financial Asset, or Group) Customer Profile Integration SOAP Process or the UCM Batch Process workflow 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 publishing of the message to the registered systems. Real-time, asynchronous, publishing increases the performance for both the UCM (Organization, Person, Financial Asset, or Group) Customer Profile Integration SOAP Process and the UCM Batch Process workflow.

You can enable the UCM Async Realtime Publish workflow to allow multiple view modes. For example you can use the All View mode which allows you to locate potential records which, though created by different users with different levels of visibility, are still the same record. If you do not specify All View mode, then the most restrictive visibility is applied to the record by default. For information on configuring All View mode, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

UCM Batch Process

This workflow is mainly called by the UCM Batch Manager server component (alias UCMBatchProcess) to process the queued Insert requests. It looks for queued records in the Source Data History table, and then it calls Oracle Customer Hub (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. This workflow displays each service call as individual workflow steps to allow easier debugging.

You can enable the UCM Batch Process workflow to allow multiple view modes. For example you can use the All View mode which allows you to locate potential records which, though created by different users with different levels of visibility, are still the same record. If you do not specify All View mode, then the most restrictive visibility is applied to the record by default. For information on configuring All View mode, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

UCM Contact Data Decay Process

This workflow calls HaleyAuthority decay business rules to evaluate the decay scores of a contact record and raises an alarm when the decay score falls below a preset threshold. It runs in batch mode using the Workflow Process Batch Manager server component. You can use search specifications to limit the number of records that are evaluated.

UCM Daily Publish

This workflow calls the UCM Publish Subscribe business service's Publish Method to work in batch mode. It picks up the updates and new records that each external system (whose Publish Frequency value equals Daily Batch) is configured to receive within a given (start and end) date-time period after the last published date and time.

The user property, ReloadSystemsCounter-10 (default value), is used by the UCM Publish Subscribe business service in batch mode to reload the system configuration information after the set number of iterations.

You can enable the UCM Daily Publish workflow to allow multiple view modes. For example you can use the All View mode which allows you to locate potential records which, though created by different users with different levels of visibility, are still the same record. If you do not specify All View mode, then the most restrictive visibility is applied to the record by default. For information on configuring All View mode, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

UCM Financial Asset Customer Profile Integration SOAP Process

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

You can enable the UCM Financial Asset Customer Profile Integration SOAP Process workflow to allow multiple view modes. For example you can use the All View mode which allows you to locate potential records which, though created by different users with different levels of visibility, are still the same record. If you do not specify All View mode, then the most restrictive visibility is applied to the record by default. For information on configuring All View mode, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

UCM Group Customer Profile Integration SOAP Process

This workflow receives data as a Simple Object Access Protocol (SOAP) message, which differs from the UCM Batch Process workflow in that it does not go through the SDH tables. It also handles errors.

UCM Import File Processing

This workflow is called from the UCM List Import Process to read and process the text format of the import file.

UCM Master List Import Process

This workflow controls the entire UCM List Import process. The processes include:

- Reading data from import files to create batch records Source Data and History (SDH) tables
- Validating the records in the SDH tables
- Invoking the UCM Batch Process to move records into the base tables

UCM Organization Customer Profile Integration SOAP Process

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

You can enable the UCM Organization Customer Profile Integration SOAP Process workflow to allow multiple view modes. For example you can use the All View mode which allows you to locate potential records which, though created by different users with different levels of visibility, are still the same record. If you do not specify All View mode, then the most restrictive visibility is applied to the record by default. For information on configuring All View mode, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

You can also add additional business logic or processing to the workflow to occur in the UCM Transaction Manager after an inbound XML message is received and processed, but before it is committed. For more information, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network

UCM Organization Hierarchy Management Integration SOAP Process

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

UCM Person Customer Profile Integration SOAP Process

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

You can enable the UCM Person Customer Profile Integration SOAP Process workflow to allow multiple view modes. For example you can use the All View mode which allows you to locate potential records which, though created by different users with different levels of visibility, are still the same record. If you do not specify All View mode, then the most restrictive visibility is applied to the record by default. For information on configuring All View mode, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

You can also add additional business logic or processing to the workflow to occur in the UCM Transaction Manager after an inbound XML message is received and processed, but before it is committed. For more information, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network

UCM Process Merge Request

This workflow validates and processes requests to merge two best version UCM records. The request might come from a regular merge or guided merge, performed in the duplicate resolution views or merge request views. You can set the process properties to enable UCM survivorship or UCM publish based on business requirements.

About Oracle Customer Hub (UCM) Server Components

Oracle Customer Hub (UCM) has one server component group and three server components to administer its functionality. The name of the component group is Oracle Universal Customer Master. The following table displays the three server components:

Server Component	Alias	Description
UCM Object Manager	UCMObjMgr	Indicates an interactive-mode server component. UCM Object Manager is the application object manager that administers all requests and data management tasks for Oracle Customer Hub (UCM).
UCM Batch Manager	UCMBatchProcess	Indicates a batch-mode server component that manages batch invocation of Customer Data Management for account and contact insert requests that have been queued in the source data history table. This server component uses the UCM Batch Process workflow.
UCM Batch Publish Subscribe	UCMBatchPubSub	Indicates a batch-mode server component that manages the batch publishing of new and updated records for subscribing external systems.

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

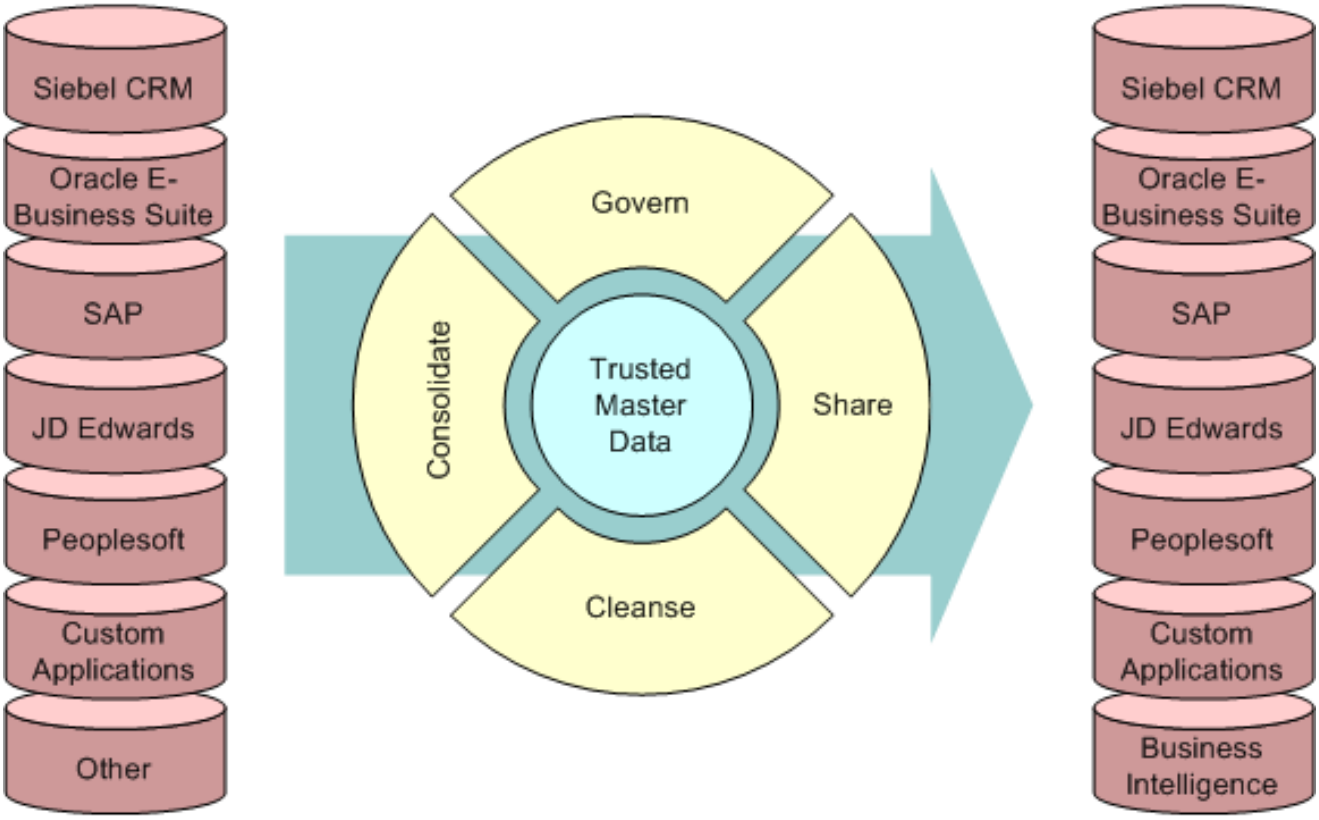
About the Universally Unique Identifier

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

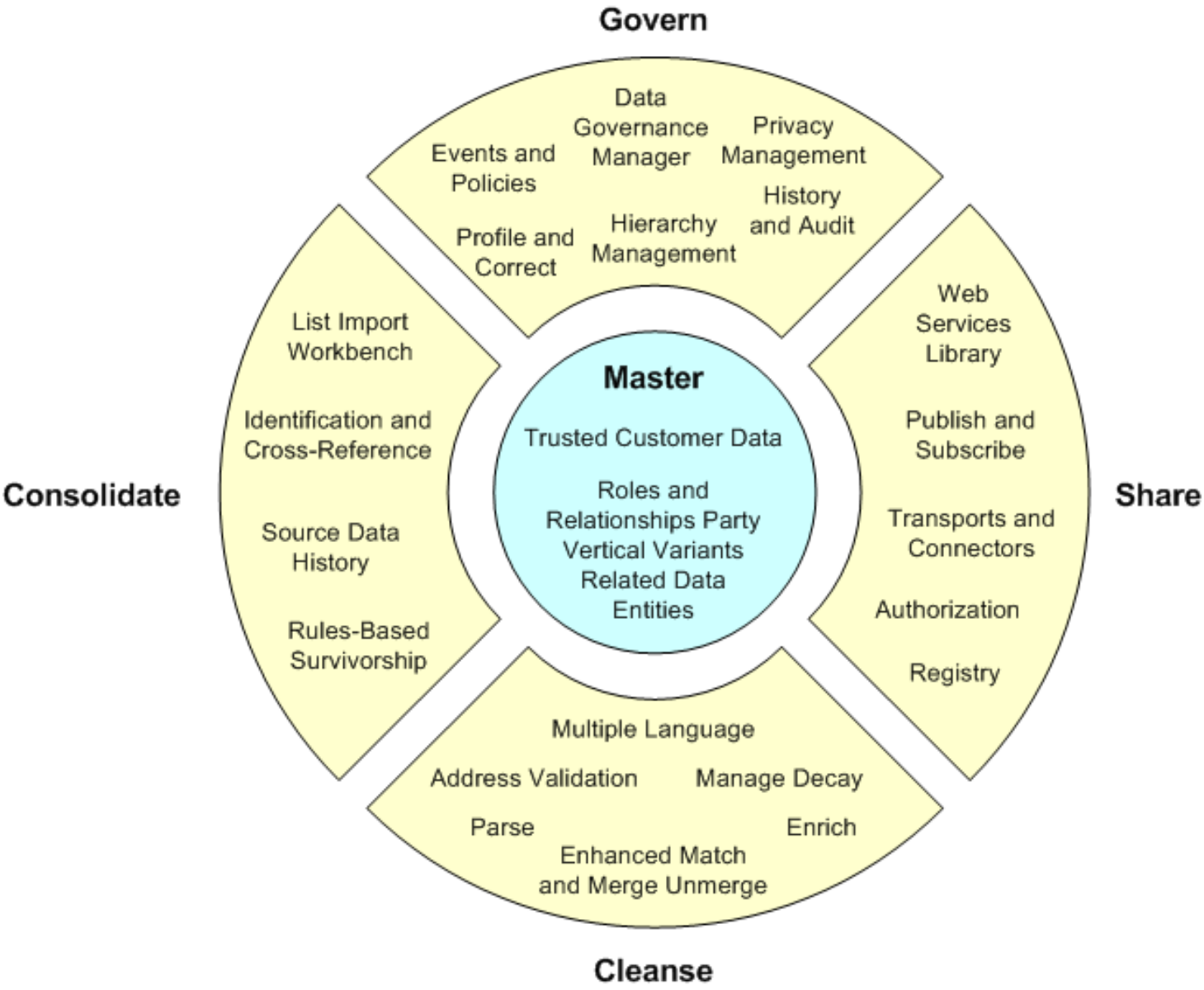
About Oracle Customer Hub (UCM) Concepts

Oracle Customer Hub (UCM) is configured to store a clean and unified profile for enterprise customer data. Oracle Customer Hub (UCM) is the primary Master Data Applications product and serves as the example for the configuration and administration tasks that are documented in this guide.

The following figure illustrates the way in which Oracle Customer Hub (UCM) consolidates customer data collected from various systems into a single, mastered collection of data, from which subscribing applications then might draw.



In addition to storing the primary set of customer data for an enterprise, Oracle Customer Hub (UCM) also includes many features to cleanse, evaluate, publish, store, and manage this customer data. The following figure illustrates the five areas of functionality for Oracle Customer Hub (UCM).



As shown in this image, the five areas of Oracle Customer Hub (UCM) functionality are as follows:

- **Master.** Trusted Customer Data, Roles and Relationships, Party, Vertical Variants, Related Data Entities.
- **Govern.** Events and Policies, Profile and Correct, Hierarchy Management, Data Governance Manager, Privacy Management, History and Audit.
- **Share.** Web Services Library, Publish and Subscribe, Transports and Connectors, Authorization, Registry.
- **Cleanse.** Multiply Language, Address Validation, Parse, Enhanced Match and Merge Unmerge, Manage Decay, Enrich.
- **Consolidate.** List Import Workbench, Identification and Cross-Reference, Source Data History, Rules-Based Survivorship.

Oracle Customer Hub (UCM) Concepts

The following topic explains the five areas of Oracle Customer Hub (UCM) functionality.

Consolidate

Use the List Import workbench to perform bulk import operations on account and contact records. Use identification cross-reference, source data history, and rules-based survivorship to enhance data quality.

Master

Maintain trusted customer data, allowing for roles and relationships. Maintain related data for master records, as well as for entities.

Cleanse

Use the Data Cleansing Engine with multilanguage capability to standardize and validate account and contact data. Use the data-decay functionality of Oracle Customer Hub (UCM) to manage data decay, parse and enrich data with enhanced merge and unmerge functionality.

Share

Use a Web services library to execute UCM operations and to configure publish-and-subscribe functionality. This functionality does the following:

- Determines how external systems receive customer data
- Exchanges data with external applications using transports and connectors for both synchronous and asynchronous communication

Govern

Govern customer data through Oracle Data Governance Manager and Oracle's Hyperion Data Relationship Management for Customer Hub solution. It enables you to do the following:

- Maintain an audit trail and history of records.
- Profile and correct customer data.
- Maintain events and enforce policies.
- Enforce privacy-management rules.

The basic features of Oracle Customer Hub (UCM) are described in the following table.

Feature	Description
Consolidate: List Import	Oracle Customer Hub (UCM) List Import function enables you to perform bulk import operations on account and contact records from external systems. For more information, see *Configuring the List Import Function*.
Consolidate: Source Data History	The Source Data History (SDH) tables maintain a record of data transactions between Oracle Customer Hub (UCM) and registered external systems. For more information on SDH tables, see *Source Data History Tables*.

Feature	Description
Consolidate: Cross-referencing	Cross-referencing allows the identification of customer data in external systems to be saved in Oracle Customer Hub (UCM), allowing a one-to-many mapping of this data. For more information on cross-referencing, see *About Cross-Referencing Parent and Child Records with External Systems*.
Consolidate: Survivorship	The survivorship feature provides a rules-based means to automate the quality of the master customer data. This data is compared to its source and age to determine whether to maintain or update customer data. For more information on survivorship, see *About Oracle Customer Hub (UCM) Survivorship Rules*.
Cleanse: Duplicates: Identification and Resolution	Oracle Customer Hub (UCM) contains functionality, such as suspect matching, merge and unmerge and Guided Merge to allow you to resolve issues with record duplication. For more information, see *Configuring Oracle Customer Hub (UCM) to Resolve Duplicate Records*.
Cleanse: UCM Data Cleansing and Matching	Oracle Customer Hub (UCM) supports data cleansing and data Matching using Oracle Data Quality Matching and Cleansing server. For more information on configuring UCM data cleansing and matching, see the following: • *Configuring the MDM Foundation and Workflows* • *Siebel Data Quality Administration Guide*
Cleanse: Data Decay Management	Data Decay Management enables you to manage information that has become degraded or obsolete. Metrics allow data decay to be managed on several different levels. For more information on data decay management, see *Configuring Data Decay Functionality in Oracle Customer Hub (UCM)*.
Share: Oracle Customer Hub (UCM) Web Services	For more information on Oracle Customer Hub (UCM) Web services, see *Oracle Customer Hub (UCM) and Data Governance Manager Web Services Reference*.
Share: Publish and Subscribe	Publish and subscribe functionality determines the details on how external systems receive customer data updates from Oracle Customer Hub. For more information on publish and subscribe functionality, see *System Publish and Subscribe Operation Modes* and *Configuring System Publish and Subscribe for Oracle Customer Hub (UCM)*.
Govern: Oracle Data Governance Manager	Oracle Data Governance Manager allows data stewards to operate and manage all Oracle Customer Hub (UCM) operations, such as data consolidation, master data creation, data cleansing, and data sharing between subscribing applications. For more information, see *Using Oracle Data Governance Manager*.
Govern: Privacy Management	Privacy Management provides an infrastructure that allows companies to implement privacy rules for their customer data. For more information, see *Configuring Oracle Customer Hub (UCM) Privacy Management*.
Govern: Workflows and Configuring the MDM Foundation	For information on workflows and configuring workflows, see *Configuring the MDM Foundation and Workflows*.
Oracle Higher Education Constituent Hub	Oracle Higher Education Constituent Hub provides a means to maintain accurate, single-source control of all the disparate constituent data in your enterprise. For more information, see *Configuring Oracle Higher Education Constituent Hub*.

Scenario for Deploying Oracle Customer Hub (UCM)

This topic describes how Oracle Customer Hub (UCM) might be deployed by a financial institution. You might use Oracle Customer Hub (UCM) differently, depending on your business model.

- *Deploying Oracle Customer Hub (UCM)*
- *Deploying Oracle Customer Hub (UCM) Using Oracle Application Integration Architecture*

Deploying Oracle Customer Hub (UCM)

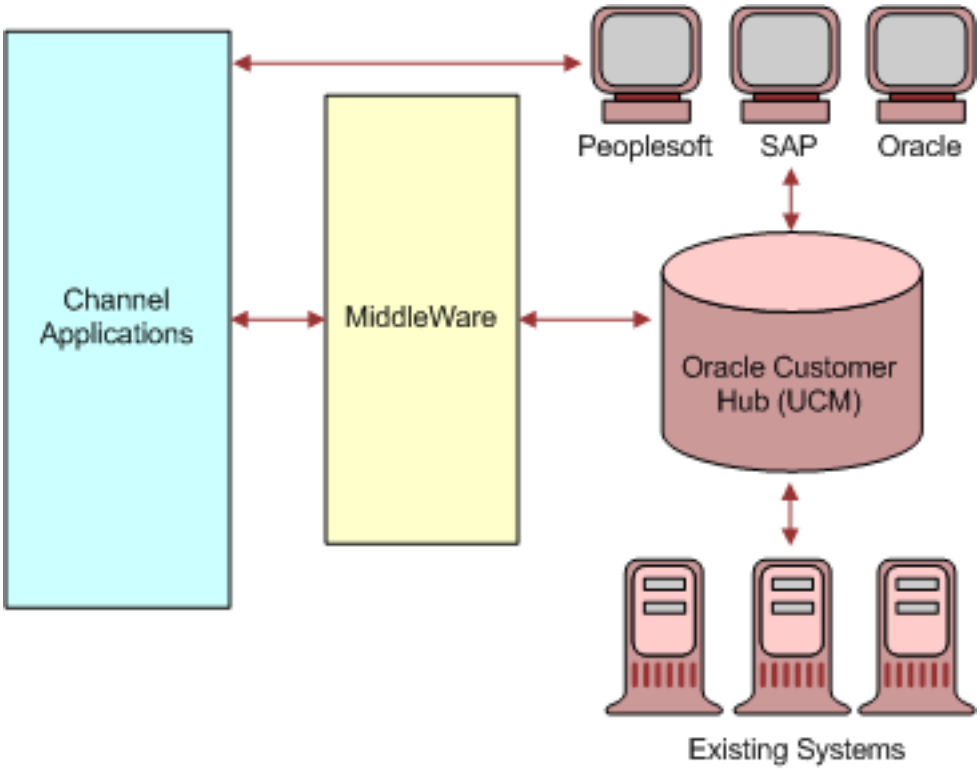
A large, successful financial institution has evolved into a national enterprise with offices and business units in several locations across the country. Its core banking systems include client deposits, loans, and mortgages. It also offers credit card services and an investment banking service. Through 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 (Human Resources departments and Accounting departments, for example) use 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 Oracle Customer Hub (UCM) as a key component in the synchronization of customer-profile information across customer-related applications, including back-office applications.

In this scenario, Oracle Customer Hub (UCM) operates as a stand-alone application in an application network. Oracle Customer Hub (UCM) stores, synchronizes, and reconciles customer data for the financial organization. See the following figure for an example of this deployment.



Deploying Oracle Customer Hub (UCM) Using Oracle Application Integration Architecture

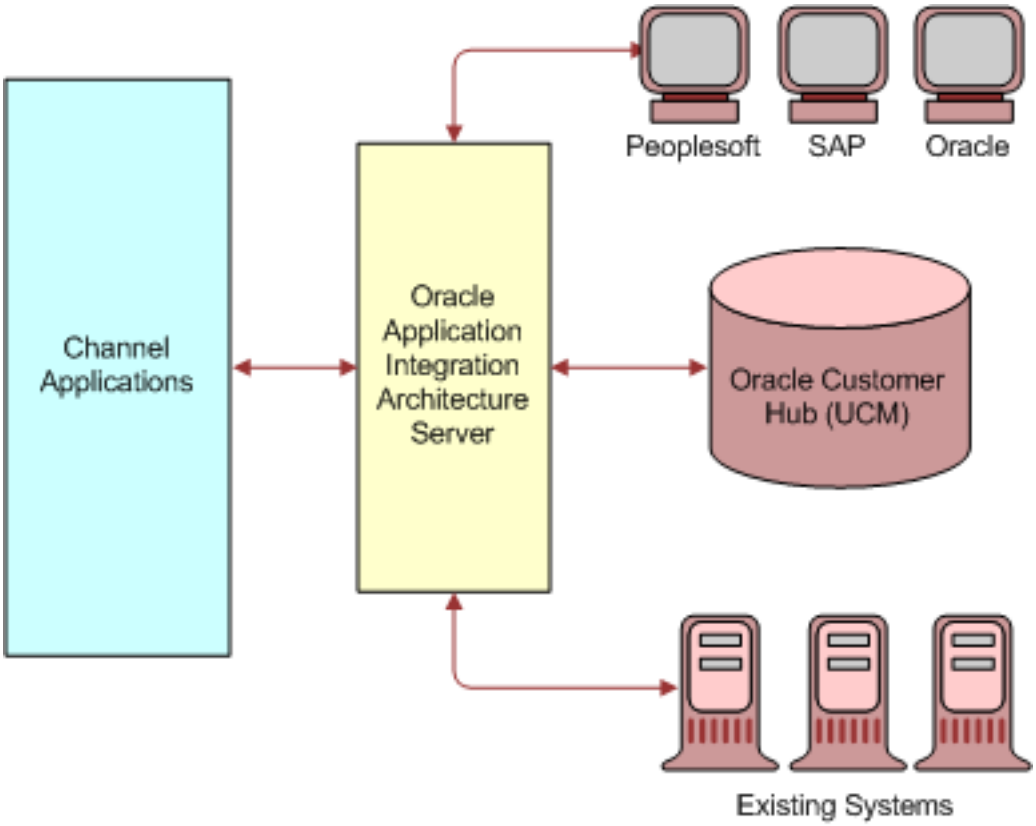
The deployment of Oracle Customer Hub (UCM) Applications uses several prebuilt connectors along with the full Siebel Enterprise Application Integration module functionality to integrate and consolidate customer information across back-office systems. For more information about this deployment, see *Oracle Customer Hub (UCM) Prebuilt Business Services*.

Oracle Master Data Management Applications use Oracle Application Integration Architecture framework to synchronize account, contact, prospect, and household data across disparate systems. Each application on Oracle Application Integration Architecture can act as a source of new and updated customer information and can also receive new and updated information from other applications.

Oracle Application Integration Architecture can also synchronize customer information between Oracle Customer Hub (UCM) and Siebel Business Applications (including previous versions). The current Oracle Application Integration Architecture business processes are used primarily for scenarios in which multiple applications store a copy of the customer profile and require Oracle Customer Hub (UCM) to act as the primary registrar to determine the validity of the new and updated customer information. The multiple applications can include Siebel Business Applications, back-office systems, and legacy applications.

Oracle Application Integration Architecture provides a reusable integration solution with Oracle Customer Hub.

In this scenario, shown in the following figure, Oracle Customer Hub (UCM) operates as a stand-alone application as in the figure in the topic *Deploying Oracle Customer Hub (UCM)* but is deployed with Oracle Application Integration Architecture Server.


Peoplesoft
SAP
Oracle
Channel Applications
Oracle Application Integration Architecture Server
Oracle Customer Hub (UCM)
Existing Systems

4 Configuring the List Import Function

Configuring the List Import Function

This chapter describes how to import and manage lists of accounts and contacts from supported applications for use in Oracle Customer Hub (UCM). It includes the following topics:

- *About List Management in Oracle Customer Hub (UCM)*
- *Scenarios for Using the List Import Function with Oracle Customer Hub (UCM)*
- *Configuring List Import Function*
- *Administering the List Import Function*

About List Management in Oracle Customer Hub (UCM)

Oracle Customer Hub (UCM) allows the business user to perform bulk import operations on account and contact records from external systems. The configurable process allows the business user to import from the following file formats: delimited text files (such as CSV, tab delimited, fixed width, or other delimiter) and XML files. The automated process consists of two steps:

- A list import into SDH (Source Data History) tables.
- A batch process that calls existing Oracle Customer Hub (UCM) services for data cleansing, data matching, and survivorship. These processes identify the best version record before committing the record to a base table. The best version record can then be published to all subscribing systems.

Oracle Customer Hub (UCM) Data Steward Functions

The data steward performs the following functions with Oracle Customer Hub (UCM) data:

- Imports data from the flat files into staging tables (SDH tables).

  If an error occurs when importing files into SDH tables, then the data steward consults the Import-Exceptions view for a detailed view of the records that failed the import. From this view the data steward corrects the data in the flat file and reimports the remaining records.
- Uses the UCM Batch Process workflow to move records from the staging tables to the base tables.

  If an error occurs, such as records failing validation, or a batch-mode process cannot be completed, then the data steward consults the Incomplete Transaction view, corrects the problem, and starts a rerun of the batch process to include the failed records.

Scenarios for Using the List Import Function with Oracle Customer Hub (UCM)

This topic gives the following examples of how the List Import function might be used with Oracle Customer Hub (UCM). You might use Oracle Customer Hub (UCM) differently depending on your business model.

- *Importing and Cleansing a Flat File*
- *Verifying Import and Updating Errors*

Importing and Cleansing a Flat File

A branch of a national bank's account-opening system sends a flat file containing a given days' customer and account records for create and update. The originating system is a registered source system for Oracle Customer Hub.

A business user takes the flat file and imports it into Oracle Customer Hub, using the List Import function. Because this file is to be processed daily, the user automates the import process to take place at 1:00 AM every day. When defining the scheduling process, the user uploads the file (a delimited text file) and selects which type of data included in the document: data intended for the Account or Contact objects. This action displays a predefined set of Business Component fields that allows the user to map the columns of the file.

Because this file is in a standard layout, the user selects a predefined mapping and starts the import process. The import process validates the format of the data. After the import process is complete, an IT user runs a server task to run the UCM Batch Process workflow in Oracle Customer Hub. All the updates are propagated on the base tables as defined by the survivorship rules predefined in Oracle Customer Hub. All imported records are also cleansed and matched.

Verifying Import and Updating Errors

Following the successful import of a flat file, a data steward at the same bank branch verifies the import process and identifies the records that were deleted because of errors during the UCM Batch Process workflow. These views show all the records that have failed the import process with a status set to Incomplete. The data steward corrects the data from the UI and changes the type of the record from Incomplete to Batch. After the type of the record is changed to Batch, the Run UCM Batch button is enabled. The data steward clicks the Run UCM Batch button to rerun the UCM Batch Process for the corrected records. For more information about importing flat files, see *Importing and Cleansing a Flat File*.

Configuring List Import Function

Use the following topics to configure List Import function for Oracle Customer Hub (UCM):

- *Creating and Scheduling Import Jobs*
- *Setting List Import System Preferences*
- *Modifying Process Properties for UCM Batch Process Workflow*

- *Configuring Import Field Mapping*
- *Configuring Additional Child Objects for the List Import Function*
- *About the ConsolidateListImportService*

About List Import Settings

You configure single import jobs or recurrent import jobs in the Oracle Customer Hub Administration, Import Screen. From this screen you perform import operations on contact and account records.

The import settings allow you to set the import options, including mapping settings, schedule the frequency of import operations, view the status of an import job, view processed files, and view and change any records that have failed during the import.

About the ConsolidateListImportService

The List Import function serves as a standalone Web service invoked from the Data Governance Management module. For more information, see *ConsolidateListImportService*.

Setting List Import System Preferences

You set the system preferences for the List Import operations from the System Preferences screen as shown in the following task.

To set List Import system preferences

1. Navigate to the Administration - Application screen, then System Preferences.
2. Perform the following query: UCM*.
3. In the System Preference Value field choose TRUE or FALSE to set each preference.
   The following information lists the system preferences for List Import function.

System Preference	Value	Description
UCM Import: Enable Validation	True or False	Optional. If set to True, then the validation process is automatically run after the records are imported into Oracle Customer Hub (UCM) staging (SDH) tables. The validation process checks if any required fields specified in the Import Mapping are missing in the imported records. Any record that fails the validation process has its UCM type code changed from Batch to Incomplete, and the validation status is inserted in the Error Message field.
UCM Import: Enable Launch BDM	True or False	Optional. If set to True, then the UCM Batch Process is automatically run after the records are imported into Oracle Customer Hub (UCM) staging (SDH) tables. This process moves records in staging to the base tables. You can also start the UCM Batch Process as a separate server task outside of the List Import function in the Server Administration screen.

Modifying Process Properties for UCM Batch Process Workflow

You can modify the process properties of the UCM Batch Process workflow to change the default behavior of this process during List Import. The following information describes the relevant process properties for the UCM Batch Process workflow.

Process Property	Default String Value	Description
OverrideServerCompParams	false	Overrides the server component parameters. This parameter must be set to false or UCMBatchProcess server tasks will use the workflow parameters and not the server task input parameters. When UCM Import: Launch BDM is enabled, the List Import function sets it to true so that the workflow parameters will be taken.
Pub/Sub	false	Enables publish and subscribe.
TurnOnCDMCleanse	false	If this flag is set to true, then Oracle Customer Hub (UCM) invokes Oracle Data Quality Manager to cleanse the records, standardize the record name, and verify the addresses before continuing to the next step. **Note:** CDM (Customer Data Management) Cleanse is performed before the CDM Match step.
TurnOnCDMExactMatch	false	If this flag is set to true, then Oracle Customer Hub (UCM) invokes Exact Match to query the base table for the record. Depending on the way it is configured, Exact Match can use either a Cross Reference Table, a Query By Example, or a Query By User Key. If Exact Match does not find the match, then Oracle Customer Hub (UCM) uses DQ Match to find the record, provided that DQ Match is enabled; that is, the TurnOnCDMMatch process property for the workflow is also set to true.
TurnOnCDMMatch	false	If this flag is set to true, then Oracle Customer Hub (UCM) invokes DQ Match, using a third-party matching engine.
TurnOnSE	false	If this flag is set to true, then Oracle Customer Hub (UCM) invokes the Survivorship Engine when merging (or updating) records.

Use the following task to change the default value of a workflow process property.

To change the default value of a workflow process property

1. In the Siebel Tools Object Explorer, click Workflow Process.
2. In the Workflow Processes view, query for the workflow process that you want to modify, for example, UCM Batch Process.
3. Lock the project or object that you are working on.
4. In the Object Explorer, click WF Process Prop.

5. In the WF Process Props view, select a process property that you want to modify.
6. In the Default String field, change the default value.
7. Step off the record to save it.

Configuring Import Field Mapping

You can configure additional Business Component User Properties to allow users to map to additional fields in the Account or Contact SDH Business Component. The format for any user properties that you add to either the Account or Contact SDH Business Component is as follows:

- Name: Map:[Object].[Field]
- Value: [SDH Business Component Name].[Business Component Field name]

For example, the following information lists the UCM Account Source Data and History business component has the following user properties.

User Property	Value
Map:Account.Name	UCM Account Source Data and History.Name
Map:Address.City	UCM Account Address Source Data and History.City
Map:Contact.Job Title	UCM Contact Source Data and History.Job Title

Use the following task to add a new user property.

To add a new user property

1. In the Siebel Tools Object Explorer, click Business Component.
2. In the Business Components screen, query for the business component that you want, for instance, UCM Account Source Data and History.
3. Lock the project or object that you are working on.
4. In the Object Explorer, click Business Component User Property.
5. In the Business Component User Property view, right– click, and select New Record.
6. In the Name field enter the User Property name, for instance, Map:Account.Name.
7. In the Value field, enter the business component name, then a period, then the field within that business component, for instance, UCM Account Source Data and History.Name.
8. Step off the record to save it.

Configuring Additional Child Objects for the List Import Function

If you have configured child SDH objects and must import these child objects with the Account or Contact parent record, then you must perform the following configuration tasks. For a list of supported child objects, see *Supported Child Objects for List Import*.

To configure additional child objects for import

1. In Siebel Tools, query for the child SDH Business Component.

   For example, query for UCM Account Address Source Data and History.
2. Lock the object or project.
3. In the Business Component User Properties view, add a Business Component User Property with the following name and value:
   - Name: Parent SDH ID Field Name
   - Value: The field that contains its parent SDH ID
4. In the parent SDH Business Component, add the additional field mapping for the child object.

   For more information on mapping, see *Configuring Import Field Mapping*.
5. In the parent SDH Business Component, for example 'UCM Account Source Data and History' or 'UCM Contact Source Data and History' add the new Business Component User Property to specify the child object Business Component for validation. The format is as follows:
   - Name: Validation:[Object Name]
   - Value: Object Business Component Name

   The following table lists the UCM Account Source Data and History Business Component validation user properties.

User Property	Value
Validate:Account	Account.
Validate:Address	CUT Address
Validate:Contact	Contact

The following table lists the UCM Contact Source Data and History Business Component validation user properties.

User Property	Value
Validate:Account	Account.
Validate:Address	Personal Address.
Validate:Contact	Contact.
Validate:SM Profile	SM Contact Social Profile.

User Property	Value

6. For the Import Business Object, either UCM Account Import or UCM Contact Import, add the new child SDH Business Component and the corresponding link to the parent SDH Business Component.

   For example, if you are configuring new child objects for the Account business object, modify the UCM Account Import Business Object. If you are configuring for the Contact business object, then modify the UCM Contact Import business object.
7. In the UCM File Manager Service Business Service, add a new Business Service user property for the new child SDH Business Component. For example, since Account Import has two child objects (Address and Contact) you must add two Business Service user properties:
   - For Account Import Staging Child Buscomp 1, add the child record, UCM Account Address Source Data and History.
   - For Account Import Staging Child Buscomp 2, add the child record, UCM Contact Source Data and History.

Supported Child Objects for List Import

The following information lists the child objects that are supported out of the box for List Import functions.

Parent Object	Child Object	Business Component Name
Account	• Contact • Address	• Contact • CUT Address
Contact	• Account • Address • Social Media Profile	• Account • Personal Address • SM Contact Social Profile

Administering the List Import Function

Use the following topics to administer the List Import functionality for Oracle Customer Hub (UCM):

- *Creating and Scheduling Import Jobs*
- *Working with Import Job Data Mapping*
- *Viewing the Status of an Import Job*
- *Viewing Processed Import Files*
- *Viewing Import Exception Files*
- *Viewing and Correcting Incomplete Import Transactions*

Creating and Scheduling Import Jobs

Each import job uses a reusable import mapping between the columns of the input data and the columns of the Oracle Customer Hub SDH tables. If you are performing an import operation and have not previously configured table mappings, then see *Working with Import Job Data Mapping*.

Oracle Customer Hub (UCM) can be configured to ignore certain fields and allow field updates while checking for duplicate records. After the job has been defined with import mappings, the job can be scheduled to run. Jobs can be executed on a recurring basis. You can monitor an activated job for exceptions and errors. For more information monitoring transactions, see *Viewing and Correcting Incomplete Import Transactions*. Use the following task to schedule a list import operation.

To schedule a list import operation

1. Navigate to the Administration - Universal Customer Master screen, then the Import view.
2. In the Import Jobs screen, query for an existing job, or click New to create a new one.
3. To create a new job, click the Select button in the Source field.

   The Systems window appears.
4. Click the system from which you want to import the batch file, for instance, AIA.

   The following table lists some of the fields for the new record.

Field	Comments
Job Name	In the text field, provide a unique job name.
Main Object	Select the object that the list import job will be updating: either the Account or Contact object.
Description	In the text field, describe the job.
Importing Mapping	Click the Select button, then in the window, select the Import Mapping template. The import mapping are saved definitions of mapped fields in a chosen business component. Select from the saved definitions, or create a new definition of the mapped fields. For more information on mapping, see *Working with Import Job Data Mapping*.
Active	A check appears in this field after you activate the import job. To activate the import job, click the Activate button. To stop the job, click the Deactivate button.

5. Click the Job Setup View tab to further define details about the source file that you will import. The following table lists the fields in the Source File Details area.

Field	Comments
File Selection	Specifies the source file. Choose one of the following values:

Field	Comments
	○ **Use exact file name.** With this option, the File Name field on the form becomes editable. ○ **Use most recent file.** Allows for recurring jobs.
File Name	Click the Select button to browse to the source file or to enter a URL.
Data Type	Click the data type. The options are: comma separated, tab delimited, other delimiter, fixed width, and XML.
Delimiter	If the data type of the source file is Other Delimiter, then you must enter a character, otherwise this field is not editable. The tab delimited and CSV data types are automatically handled.
Header Row Included	Select this check box if the file includes a header row.
Schedule Mode	Click the list, and choose a value. The values include: One time (manually started), One time (scheduled), and Repeating. **Note:** You must enter a value in the Schedule Mode field before activating a job.
Start Date	Click the Select button, then chose a date and time from the Calendar window, and click Save.
Repeat Frequency	Click the Select button to enter a numeric value for the frequency of the job being run. You set the unit of frequency in the next field.
Repeat Unit	Choose the repeat unit. The values include: hours, days, weeks, or months.
End Date	Click the Select button, then choose an end date and time from the Calendar widow, and click Save.

6. In the Input Arguments view, click New if you want to add new input arguments.
7. Click Activate to make the import job active.

Working with Import Job Data Mapping

Before you import a new file, you must map the fields in the imported list to the fields in either the Contact or the Account business component. When you do this mapping, the values in the list fields are transferred into the correct fields in the record.

There are two ways to create a new import mapping template:

- To create a new import mapping template, see the first procedure in this topic.
- If you have already created an import job and attached the import file to the job, then see second procedure in this topic.

Use the following task to create a new import mapping template.

To create a new import mapping template

1. Navigate to the Administration - Universal Customer Master screen, then the Import Mappings view.
2. In the Import Mappings screen, query for an existing Import Mappings template, or click New to create a new one.
3. Enter the Reference Name and Main Object (either the Account or Contact object) for this Import Mapping template.
4. Click on the New button to create new Field Mapping.
5. For each entry, click the Select button in the Field Name field.
   The Import Mappings picklist appears listing the import object and the field in that object to which the data is mapped.
6. Choose an entry, then click OK.
7. Repeat steps 4 through 6 for each field you want to map to.
   Use the following task to map the fields of an import job.

To map fields for an import job

1. After successfully creating an Import Job as outlined in *Creating and Scheduling Import Jobs*, click the Data Mapping tab.
2. In the Data Mapping view, click Refresh Mapping.
   The provided input file is read and the sample input data is displayed.
3. For each entry, click in Input Field, then click the Select button.
   The Import Mappings picklist appears listing the import object and the field in that object to which the data is mapped.
4. Choose an entry and click OK.
5. Repeat for each field that you want to map to.

Viewing the Status of an Import Job

Use the following task to view the status of an import job.

To view the status of an import job

1. Navigate to the Administration - Universal Customer Master screen, then the Import view.
2. In the Import Jobs screen, select the job, then click the Job Status View tab.
   The status of the import is displayed in the Job Status view.
3. Navigate to the Task Summary view to access a log file of the import transaction or a task summary of the transaction.
4. To switch between the view options, click the list, and select either Task Log or Task Summary.

Viewing Processed Import Files

To view import files processed during an import operation, use the Files Processed view on the Import Jobs screen as shown in the following task.

To view processed import files

1. Navigate to the Administration - Universal Customer Master screen, then the Import view.
2. In the Import Jobs screen, select the job, then click the Files Processed View tab.

   The Files Processed view displays all processed files for the import jobs.

Viewing Import Exception Files

To view exception files from an import operation, use the Exceptions view on the Import Jobs screen as shown in the following task.

To view import exception files

1. Navigate to the Administration - Universal Customer Master screen, then the Import view.
2. In the Import Jobs screen, select the job, then click the Exceptions View tab.

   The Exceptions view displays all excepted files from the import job.

Viewing and Correcting Incomplete Import Transactions

To view or correct incomplete import transactions, use the Incomplete Transactions screen. This screen enables you to view all account and contact transactions that failed the import validation or failed during the UCM batch process. You can view information about the record, as well as the specific error message. Use the following task to view and, if necessary, correct incomplete import transactions.

To view and correct incomplete import transactions

1. Navigate to the Administration - Universal Customer Master screen, then the Incomplete Transactions view.
2. Click the list, and select either Incomplete Accounts or Incomplete Contacts.
3. In the chosen view, find the job record.
4. For incomplete account records, click the Account link in the record.

   The Incomplete Account Detail view appears, displaying the relevant data associated with the chosen account record, including its Address and Contact child objects.
5. For incomplete contact records, click the Last Name link.

   The Incomplete Contact Detail view appears, displaying the relevant data associated with the chosen contact record, including its Address and Account child objects.
6. Make the necessary corrections, and change the Record Type from Incomplete to Batch, then step off the record to save it, or choose Save Record from the application level menu.

7. Click either Incomplete Contact Details or Incomplete Account Details on the thread bar to return to the previous view.
8. Click the Run UCM Batch button to rerun the UCM Batch Process for the selected record.

> **Note:** You can select multiple records in this view.

5 Configuring Source Data History and Cross-Reference Functionality

Configuring Source Data History and Cross-Reference Functionality

This chapter describes how to configure and administer source data history and cross-reference functionality for Oracle Master Data Management Applications. It includes the following topics:

- *Configuring Source Data History and Cross-Reference Functionality for Oracle Customer Hub*
- *Administering Source Data and Cross-Reference Functionality for Oracle Customer Hub (UCM)*

Configuring Source Data History and Cross-Reference Functionality for Oracle Customer Hub

Perform the following tasks to configure duplicate resolution functionality for Oracle Customer Hub (UCM):

- *Loading Data into Oracle Customer Hub (UCM) and Updating External Systems*
- *Running Data Management in Batch Mode*
- *About System Preferences for Oracle Customer Hub (UCM)*
- *System Publish and Subscribe Operation Modes*
- *Configuring System Publish and Subscribe for Oracle Customer Hub (UCM)*

Loading Data into Oracle Customer Hub (UCM) and Updating External Systems

After installing Oracle Customer Hub (UCM), the initial customer data must be loaded into the database. To prepare for this process, it is recommended that you do the following:

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

Use Siebel Enterprise Integration Manager (Siebel Enterprise Integration Manager) to load bulk data records into Oracle Customer Hub (UCM). For more information on Siebel Enterprise Integration Manager, see the following:

- *Siebel Enterprise Integration Manager and Integration of Oracle Customer Hub (UCM)*
- *Siebel Enterprise Integration Manager Administration Guide*

Activating Oracle Customer Hub (UCM) External Systems

After loading the customer data into Oracle Customer Hub (UCM), you must register and activate external systems that access the customer master data. For more information registering external systems, see *Registering Systems Connected to Oracle Customer Hub (UCM)*.

Running Data Management in Batch Mode

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

This component job fetches the IDs of the records and calls the batch workflow. The workflow queries, cleanses, matches, and publishes Oracle Customer Hub (UCM) records. Three match cases can occur: no match, one match, or multiple matches. The workflow calls UCM Transaction Manager and survivorship rules internally, depending upon the match case type. For background information on administering component jobs, see *Siebel System Administration Guide* . Use the following task to use UCM Data Management to schedule a batch workflow.

To run UCM Data Management in batch mode by scheduling a batch workflow

1. Navigate to the Administration - Server screen, then the Jobs view.
2. In the Jobs list, click New.

   The component job status field changes to Creating.
3. In the Component/Job field, click Select, then query in the window for UCMBatchProcess.

   UCMBatchProcess is the short name for UCM Batch Manager.
4. In the Job Detail view, enter data in the appropriate fields to describe the start time and so on of the component job.
5. From the menu, choose Save Record.
6. In the Job Parameters list, click New to add each of the following parameters for the component job.

Parameter Name	Value
UCM Batch Object Type	Select Contact or Account.
UCM Batch Size	Select the page size of the batch job. The default value is 10.
UCM Data Management Cleanse Flag	Make sure this value is set to True.
UCM Data Management Exact Match Flag	Make sure this value is set to True.
UCM Data Management Match Flag	Make sure this value is set to True.

Parameter Name	Value
UCM Publish/Subscribe Flag	The values are True or False.
UCM Search Specification	A search specification is required. Make sure a search specification is set on UCM Type Code, for example: `[Contact.UCM Type Code] = 'Queued-Insert'`.
UCM Sort Specification	Optional sort specification of what order the to process the records in, for example, for contact records: Created (ASCENDING), Last Name, First Name.
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 Oracle Customer Hub (UCM)

Oracle Customer Hub (UCM) system preferences control how Oracle Customer Hub (UCM) operates and what functionality is enabled in the environment. You might want to review the default settings and modify these as appropriate, depending on your business needs. For more information on setting system preferences, see *Siebel Applications Administration Guide* .

To set system preferences for Oracle Customer Hub (UCM)

1. Navigate to Administration - Application, then System Preferences.
   Query for the values listed in the following table.

System Preference	Default Value	Description
Enable CDM Cleanse 1	Account	Turns on CDM Cleanse for the Account object.
Enable CDM Cleanse 2	Contact	Turns on CDM Cleanse for the Contact object.
Enable CDM EM 1	Account	Turns on CDM Exact Match for the Account object.
Enable CDM EM 2	Contact	Turns on CDM Exact Match for the Contact object.
Enable CDM Match 1	Account	Turns on CDM Match for the Account object.
Enable CDM Match 2	Contact	Turns on CDM Match for the Contact object.

System Preference	Default Value	Description
Enable Child Cross Referencing	FALSE	Enables or disables cross-reference functionality on child records.
Enable Survivorship Engine 1	Account	Enables survivorship on the Account object.
Enable Survivorship Engine 2	Contact	Enables survivorship on the Account object.
Enable Survivorship Engine 3	Household	Enables survivorship on the Account object.
Enable Entity Survivorship	FALSE	Enables survivorship of child records.
Enable UCM Processes	FALSE	Set this value to True to enable integration with other applications and Oracle Customer Hub. For more information, see *About the Integration of Oracle Customer Hub (UCM) with Siebel Business Applications*.
UID Generation Service	FINS Teller Converter Extensions: GenerateID, and DType|GUID.	The business service name, method, and input combination to generate a unique ID value. It is used by both UCM Transaction Manager and UI Interaction service.
Unique Id: Account	Party UId	Unique ID field for account. It is used with UCM Transaction Manager and UI Interaction service.
Unique Id: Contact	Party UId	Unique ID field for contact. It is used with UCM Transaction Manager and UI Interaction service.
Unique Id: FINCORP Account	Account Number	Unique ID field for the Financial account. It is used with UCM Transaction Manager and UI Interaction service.
Unique Id: Household	Party UId	Unique ID field for household. It is used with UCM Transaction Manager and UI Interaction service.

Additions and Enhancements for Social Profile Child Object Cross Reference Functionality

The following topics describe additions and enhancements to the social profile child cross reference functionality:

- *Enhanced Screen: CIF Administration Screen*
- *New View: UCM SM Contact Social Profile Cross Reference View*
- *UCM SM Contact Social Profile Cross Reference View Web Template Items*
- *New Applet: CIF SM Contact Social Profile References List Applet*
- *List Columns for the CIF SM Contact Social Profile Attributes References List Applet*
- *New Applet: CIF SM Contact Social Profile Attributes References List Applet*
- *List Columns for the CIF SM Contact Social Profile Attributes References List Applet*

Enhanced Screen: CIF Administration Screen

The following information lists details of the modified CIF Administration screen.

New or Modified	Screen	Comments
Modified	CIF Administration Screen	There are two new views: • UCM SM Contact Social Profile Cross Reference View • UCM SM Contact Social Profile History View

New View: UCM SM Contact Social Profile Cross Reference View

The following information lists the properties of the UCM SM Contact Social Profile Cross Reference View.

Name	Web Template
Base	View Detail Multi Child

UCM SM Contact Social Profile Cross Reference View Web Template Items

The following information lists the properties of the view Web template items for UCM SM Contact Social Profile Cross Reference View.

New or Modified	View	Applet	Applet Mode
New	UCM SM Contact Social Profile Cross Reference View	Contact Form Applet	Edit
		SM Contact Social Profile List Applet	Edit List
		SM Contact Social Profile Attributes Applet	Edit List
		CIF SM Contact Social Profile References List Applet	Edit List
		CIF SM Contact Social Profile Attributes References List Applet	Edit List

New Applet: CIF SM Contact Social Profile References List Applet

The following information lists the properties of CIF SM Contact Social Profile References List Applet.

Field	Value
Title	External Social Profile IDs
Business Component	CIF SM Contact Social Profile Reference
Class	CSSFrameList

List Columns for the CIF SM Contact Social Profile References List Applet

The following information lists the properties of the list columns of CIF SM Contact Social Profile References List Applet.

Field	Display Name	HTML Type	Run Time
System Number	System Id	Field	TRUE
System Name	System Name	Field	TRUE
External Id1	External Id1	Field	TRUE
External Id2	External Id2	Field	TRUE
External Id3	External Id3	Field	TRUE
Comment	Comment	TextArea	TRUE

New Applet: CIF SM Contact Social Profile Attributes References List Applet

The following information lists the properties of CIF SM Contact Social Profile Attributes References List Applet.

Field	Value
Title	External Social Profile Attribute IDs
Business Component	CIF SM Contact Social Profile Attributes Reference
Class	CSSFrameList

List Columns for the CIF SM Contact Social Profile Attributes References List Applet

The following information lists the properties of the list columns of CIF SM Contact Social Profile Attributes References List Applet.

Field	Display Name	HTML Type	Run Time
System Number	System Id	Field	TRUE
System Name	System Name	Field	TRUE

Field	Display Name	HTML Type	Run Time
External Id1	External Id1	Field	TRUE
External Id2	External Id2	Field	TRUE
External Id3	External Id3	Field	TRUE
Comment	Comment	TextArea	TRUE

New Applets for Source Data History Functionality

The following topics describe the new and enhanced applets for the source data history functionality:

- *New View: UCM SM Contact Social Profile History View*
- *UCM SM Contact Social Profile History View Web Template Items*
- *New Applet: UCM SM Contact Social Profile Source Data and History List Applet*
- *UCM SM Contact Social Profile Source Data and History List Applet Columns*
- *New Applet: UCM SM Contact Social Profile Attribute Source Data and History List Applet*
- *UCM SM Contact Social Profile Attribute Source Data and History List Applet Columns*

New View: UCM SM Contact Social Profile History View

The following information lists the properties of the view Web template for UCM SM Contact Social Profile History View.

Name	Web Template
Base	View Detail Multi Child

UCM SM Contact Social Profile History View Web Template Items

The following information lists the properties of the view Web template items for UCM SM Contact Social Profile History View.

Business Object	View Web Template Items	Applet Mode
Contact	Contact Form Applet	Edit

Business Object	View Web Template Items	Applet Mode
	SM Contact Social Profile List Applet	Edit List
	UCM SM Contact Social Profile Source Data and History List Applet	Edit List
	SM Contact Social Profile Attributes Applet	Edit List
	UCM SM Contact Social Profile Attribute Source Data and History List Applet	Edit List

New Applet: UCM SM Contact Social Profile Source Data and History List Applet

The following information lists the properties of UCM SM Contact Social Profile Source Data and History List Applet.

Field	Value
Title	Social Profile History
Business Component	UCM SM Contact Social Profile Source Data and History
Class	CSSFrameList

UCM SM Contact Social Profile Source Data and History List Applet Columns

The following information lists the properties of the list columns of UCM SM Contact Social Profile Source Data and History List Applet.

Field	Display Name	HTML Type	Run Time
UCM Type Code	Record Type	Text	TRUE
Author	Author	Text	TRUE
Author Link	Author Link	Text	TRUE

Field	Display Name	HTML Type	Run Time
Author Id	Author Id	Text	TRUE
Community	Community	Text	TRUE
Supress Profile Attributes	Supress Attributes	CheckBox	TRUE

New Applet: UCM SM Contact Social Profile Attribute Source Data and History List Applet

The following information lists the properties of UCM SM Contact Social Profile Attribute Source Data and History List Applet.

Field	Value
Title	Social Profile Attributes History
Business Component	UCM SM Contact Social Profile Attributes Source Data and History
Class	CSSFrameList

UCM SM Contact Social Profile Attribute Source Data and History List Applet Columns

The following information lists the properties of the list columns of UCM SM Contact Social Profile Attribute Source Data and History List Applet.

Field	Display Name	HTML Type	Run Time
UCM Type Code	Record Type	Text	TRUE
Name	Name	Text	TRUE
Value	Value	Text	TRUE
Source	Source	Text	TRUE

Administering Source Data and Cross-Reference Functionality for Oracle Customer Hub (UCM)

New or Modified	Screen	Comments
New	CIF Account Address Reference Screen	

New or Modified	View	Applet
New	UCM Other Entity Rule Set Attribute Group View	UCM Other Entity Rule Set List Applet
New	UCM Other Entity Rule Set Attribute Group View	UCM Other Entity Source Confidence Level List Applet
		Parent Applet: UCM Other Entity Rule Set List Applet
New	Other Entity Rule Set Attribute Group View	UCM Other Entity Rule Set List Applet
New	UCM Other Entity Rule Set Attribute Group View	UCM Other Entity Source Confidence Level List Applet
		Parent Applet: UCM Other Entity Rule Set List Applet
New	UCM Account Detail - Address X- Ref View	Group Applet: UCM Account Address X -References List Applet

Functionality for Oracle Customer Hub (UCM)

To administer source data and cross-reference functionality in Oracle Customer Hub (UCM), perform the following tasks:

- *Cross-Referencing Parent Records with External Systems*
- *Monitoring the History of Oracle Customer Hub (UCM) Records*
- *Monitoring Source Data of Oracle Customer Hub (UCM) Records*
- *Purging Source Oracle Customer Hub (UCM) Data*
- *Registering Systems Connected to Oracle Customer Hub (UCM)*
- *Setting System Privileges for Oracle Master Data Management Applications*

About Cross-Referencing Parent and Child Records with External Systems

Cross-referencing customer data allows organizations to store the customer identification data from external systems within Oracle Customer Hub (UCM). This cross-reference permits a one-to-many mapping of customer records across multiple systems throughout the organization. In other words, one record in Oracle Customer Hub (UCM) can map to one or more records in each registered application in Oracle Customer Hub (UCM).

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 entered into Oracle Customer Hub (UCM). The Unique Customer Identification number from external systems is embedded within the <Id> tag of the request message.

Cross-referencing is supported for parent records (account, contact, and household party objects, as well as financial accounts) and child records (account-address, account-financial accounts, contact-address, contact-financial accounts, contact-emails, contact-phones, contact-names, contact-social profiles, and contact-social profile attributes). The SOAP and Batch processes automatically cross-reference parent records, and when the System Preference *Enable Child Cross Referencing* is enabled, the processes also automatically cross-reference child records. For information on this task, see *Cross-Referencing Parent Records with External Systems*. For information on cross referencing child records, see *Cross-Referencing Child Records with External Systems*.

Cross-Referencing Parent Records with External Systems

You can cross-reference parent records in Oracle Customer Hub (UCM) with external systems to store source identification data in Oracle Customer Hub (UCM). For more information on cross-referencing records, see *About Cross-Referencing Parent and Child Records with External Systems*. Use the following task to cross-reference a contact, account, household, or financial account record.

To cross-reference an contact, account, financial account or household record

1. Navigate to the Administration - Universal Customer Master screen, then either the Contacts, Accounts, Households, or Financial Account view.
2. In the Contacts, Accounts, Households, or Financial Account list, select the record of interest.
3. Select the External IDs tab for contacts, accounts, financial account, or households.

   Existing cross-references can be viewed under the child record. To add additional cross-references, see Step 4.
4. In the External IDs view, click New.
5. Enter the appropriate information in the available fields to define the external customer and the Unique Account Identification, for each contact, account, or household party, or financial account record in Oracle Customer Hub (UCM).

   Use the following table for information on each of the fields.

Field	Description
System Id	The system ID represents an identifier for an application that accesses Oracle Master Data Management Applications. Select this value from the systems entered in the System Registrations view.

Field	Description
System Name	The system name is the name of the application that accesses Oracle Customer Hub (UCM). This value is the default name when a system number is selected.
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 the 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 that references the record in Oracle Customer Hub (UCM).

Cross-Referencing Child Records with External Systems

Organizations potentially have a large number of subscribing systems connected to Oracle Customer Hub (UCM), and thus, have a need for real-time cross-reference functionality on parent and child records. For instance, in a life science deployment a physician may work at multiple locations, and be associated with different hospitals. To update a patient's address the physician must ensure that not just the parent record, but any accompanying child records are updated or created only on one hospital system rather than all subscribed systems to avoid privacy and legal implications. The Oracle Customer Hub (UCM) child cross-referencing infrastructure allows you to store child cross-referencing information for the parent Contact, or Account record. Child cross-referencing details are also published to external systems registered for Publish/Subscribe. Child objects include addresses, financial accounts, contact points (such as email, or phone), alias records, social profile and social attribute records, as well as custom child objects.

Use the following task to cross-reference an address, financial account, email, phone, alias record, social profile and social profile attribute.

To cross-reference an address, financial account, email, phone, alias, social profile, or social attribute record

1. Navigate to the Administration - Universal Customer Master screen, then either the Contacts, or Accounts view.
2. In the Contacts or Accounts list, select the parent record of interest.
3. Select the Addresses, Financial Accounts, Phones, Emails, Names, or Social Profiles tab for contacts, or select the Addresses or Financial Accounts tab for accounts.
4. In the Addresses, Financial Accounts, Phones, Emails, Names, Social Profiles, or Social Profile Attributes list, select the child record of interest.
5. In the External IDs list, click New.

6. Enter the appropriate information in the available fields to define the external customer and the Unique Account Identification, for each child record in Oracle Customer Hub (UCM).

   Use the following table for information on each of the fields.

Field	Description
System Id	The system ID represents an identifier for an application that accesses Oracle Master Data Management Applications. Select this value from the systems entered in the System Registrations view.
System Name	The system name is the name of the application that accesses Oracle Customer Hub (UCM). This value is the default name when a system number is selected.
External Id1	The External Id1 stores the unique identifier of the external system record. This value is stored in the Id field for child record in the 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 that references the record in Oracle Customer Hub (UCM).

Source Data History Tables

Oracle Customer Hub (UCM) Source Data History tables (SDH tables) contain the transactional contact and account data records pertaining to Oracle Customer Hub (UCM) and registered external systems. These tables hold incoming and historical data records and provide the content for Oracle Customer Hub (UCM) administration views. The storage of this data allows for Oracle Customer Hub (UCM) features, such as incoming duplicates resolution, merge and unmerge, list import, 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 visible from the Siebel application, such as S_PARTY, S_CONTACT, S_USER, and store the master copy of the record. The SDH tables contain a large amount of data and must be purged at regularly. For more information on this task, see *Purging Source Oracle Customer Hub (UCM) Data*.

SDH Type Codes

The following information lists the type codes for Source Data History and when they are used by Oracle Customer Hub (UCM).

Type	Description
Batch	Records loaded by Siebel Enterprise Integration Manager into the SDH tables for batch processing.
Cleansed	A cleansed record from a manual match operation, which is stored temporarily for a Promote or Link and Update operation.
Deleted	Not used.
History	The Best Version Record prior to any modifications.
Incomplete	A record fails to process completely because of an error during real-time or batch-mode processing.
Merged	A version of the victim record before it is merged and deleted.
Queued	Not used.
Queued-Insert	The insert request record was queued in SDH for batch-mode processing.
Queued-Update	The update request record was queued in SDH for batch-mode processing.
Queued-Upsert	The upsert request record was queued in SDH for batch processing.
Source	The request in its original state from an external system after it is processed in real-time.
Source-Insert	The insert request in its original state from an external system after it is processed. When the Batch workflow is run the record type is changed from from Queued-Insert to Source-Insert.
Source-Update	The update request in its original state from an external system after it is processed. When the Batch workflow is run the record type is changed from Queued-Update to Source-Update.
Source-Upsert	The upsert request in its original state from an external system after it is processed. When the Batch workflow is run the record type is changed from Queued-Upsert to Source-Upsert.
UnMerged	A version of the victim record that has been unmerged or reactivated.

Monitoring the History of Oracle Customer Hub (UCM) Records

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

For information on other Oracle Customer Hub (UCM) data management tasks, see *About Cross-Referencing Parent and Child Records with External Systems*.

Use the following task to monitor the history of Oracle Customer Hub (UCM) records.

To monitor history of Oracle Customer Hub (UCM) records

1. Navigate to the Administration - Universal Customer Master screen, then either the Contacts or Accounts view.
2. In the Contacts or Accounts list, select the record of interest.
3. Select the Historical Version tab.
4. In the Historical Version list, review the records from different source systems that make up the Customer Hub version of the record in the form.

   This list also contains the historical versions of that particular record within Oracle Customer Hub (UCM).

Monitoring Source Data of Oracle Customer Hub (UCM) Records

Monitoring the source data and history of Oracle Customer Hub (UCM) records from this view reveals a global view of the evolution of its records. For information on other Oracle Customer Hub (UCM) data management tasks, see *About Cross-Referencing Parent and Child Records with External Systems*.

Use the following task to monitor source data for contacts or accounts.

To monitor the source data for contacts or accounts

1. Navigate to the Administration - Universal Customer Master screen, then the Transaction History view.
2. Select either Accounts or Contacts.
3. In the Transaction History list, review the records that show the source data and history records for contacts or accounts.

   The content is similar to the Historical Version view except that the Historical Version view shows only source and history data for a specific Oracle Customer Hub (UCM) record. This view shows the source data and history data for all Oracle Customer Hub (UCM) records.

Purging Source Oracle Customer Hub (UCM) Data

Maintaining historical source data for Oracle Customer Hub (UCM) records leads to large amounts of information stored in the database (SDH Tables). Oracle Customer Hub (UCM) provides purge functionality to clear this data, when necessary. For information on other Oracle Customer Hub (UCM) data management tasks, see *About Cross-Referencing Parent and Child Records with External Systems*.

CAUTION: The purge data operation removes all current records in the view and cannot be undone.

Use the following task to purge source data.

To purge source data

1. Navigate to the Administration - Universal Customer Master screen, then the Transaction History view.
2. Select either Accounts or Contacts.
3. In this list, click the Purge button.

About Oracle Customer Hub (UCM) Data Management Services

Oracle Customer Hub (UCM) provides the following data-management services to cleanse, identify, and link incoming source data to the master data records:

- Account Data Management Service
- Contact Data Management Service

These services are invoked through the Siebel Enterprise Application Integration (Siebel EAI) interface and can process single or multiple records. Records updated directly within Oracle Customer Hub (UCM) also trigger similar data management services but do not trigger the survivorship rules. These services take an incoming account or contact record and do the following:

- Check the system privilege of the application submitting the record.
- Cross-reference the record if an external account ID is provided.
- Create the Universally Unique Identifier (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 that is based on the configured parameter. The default value for account records is External Account ID, and the default value for contact records is External Contact ID.
- If an exact match is not found, call the Data Matching Engine to perform fuzzy matching to identify possible duplication. The default is Account Name for account records. The default is Contact first name and last name for contact records.

> **Note:** Oracle Customer Hub (UCM) provides an embedded data matching and cleansing solution. The Siebel Data Quality module is licensed separately from Oracle Customer Hub (UCM). See *Siebel Data Quality Administration Guide* and your third-party data quality provider documentation for more information.

Depending on the matching score returned from the Data Matching Engine, Oracle Customer Hub (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. You can configure these threshold numbers using Siebel Tools. The matching score results in the following three possibilities:

- When the incoming record is matched above the upper threshold, Oracle Customer Hub (UCM) applies survivorship rules to merge the incoming record with an existing record in Oracle Customer Hub (UCM). For information on survivorship and on setting up the rules, see *Process of Creating Survivorship Rules*.
- When the incoming record is matched below the lower threshold, Oracle Customer Hub (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 Oracle Customer Hub (UCM)*.
- When the incoming record is matched between the upper-threshold number and the lower-threshold number, Oracle Customer Hub (UCM) stores the record in the deduplication table (S_UCM_DEDUP) for further investigation. Data stewards can view the pending records in the Incoming Duplicates-Account or Contact screen.

  Configure your matching threshold and the Data Matching Engine rules based on your organization's data quality standards.

6 Configuring Oracle Customer Hub (UCM) to Resolve Duplicate Records

Configuring Oracle Customer Hub (UCM) to Resolve Duplicate Records

This chapter describes how to configure Oracle Customer Hub (UCM) to resolve duplicate records through Suspect Match, Merge and Unmerge, and Guided Merge. It includes the following topics:

- *Configuring Duplicate-Resolution Functionality*
- *Administering Duplication Resolution Functionality*

Configuring Duplicate-Resolution Functionality

The following topics describe how to configure duplicate-resolution functionality for Oracle Customer Hub (UCM):

- *Configuring Oracle Customer Hub (UCM) Auto Merge and Manual Review Threshold Values*
- *About Suspect Matching*
- *Scenario for Creating a New Record with Suspect Matching*
- *Configuring Suspect Match*
- *Suspect Match Object Additions*
- *Enabling the Link and Update Feature*
- *Scenario for Using Account and Contact Merge and Unmerge*
- *Configuring Merge Request*
- *About Guided Merge for Oracle Customer Hub (UCM)*
- *Scenario for Using Guided Merge*
- *Configuring Guided Merge*
- *Additions and Enhancements for Account and Contact Merge and Unmerge Object*
- *Managing Duplicate Records from External Systems with Suspect Match*
- *Managing Duplicate Records from External Systems with Link and Update*
- *About Merge and Unmerge*
- *Merging Duplicate Records in Oracle Customer Hub(UCM)*
- *Duplicate Resolution of Suspects Within Existing Duplicates View in B2C Deployment*
- *Unmerging Previously Merged Records in Oracle Customer Hub (UCM)*

Configuring Oracle Customer Hub (UCM) Auto Merge and Manual Review Threshold Values

Configure Oracle Customer Hub (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 the Data Management screens for review. For information on manually reviewing records in the Data Management screen, see the following:

- *Managing Duplicate Records from External Systems with Suspect Match*
- *Managing Duplicate Records from External Systems with Link and Update*

For information 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 Data Quality Manager*.

Use the following task to configure thresholds for auto merge and manual review.

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

About Suspect Matching

Previously when an incoming account or contact record was flagged for manual review by the UCM data management workflow, the record was placed in Review status, and a new record was not created. This action disrupted the overall business process workflow involving Oracle Customer Hub (UCM) and other connected applications.

What Happens When an Incoming Record Has a Status of Review?

With the latest release of Oracle Customer Hub (UCM), when an incoming record is flagged for manual review a new best version record is also created, with the appropriate UUID and cross-reference. The best version record is given a status of Suspect to be differentiated from the previously reviewed records. A response message is returned to the requesting external application. The suspect record and any identified potential duplicates that are flagged by the UCM

data management workflow are stored in a duplicate results table and made available for the data steward to review in the Duplicate Account or Duplicate Contact views. The suspect record can be approved or merged in these views. The update and delete actions can be performed in the Account and Contact views.

While the record has a Suspect status, it cannot be published to any other subscribing systems. If the suspect record already exists in the external application, the Update and Delete operations on the suspect record will keep the external application informed. The Unmerge operation restores the merged record to its previous state.

If the data steward deems that the record is not a match for any of the existing records, the Suspect status is removed, and an upsert message is sent to the external applications that do not yet have the suspect record. When the suspect record is merged into another record, a message is sent to all external applications to preserve the synchronization. When the Suspect status is removed from the record, the result is sent to those applications that do not yet have the record.

What Happens When a Record Has a Status of Pending?

While this record still has the Pending status, it can be updated, deleted or merged in the UI or through an incoming message. If the update is sent from a different system and not from the originating system (through auto match), then a cross-reference must be created for the additional system. Before the Pending status is resolved, any changes to this best version record must be published to all the systems that have an associated cross-reference associated. When the pending record is identified as a potential duplicate with another nonpending best version record, it is recommended that the pending record be merged into the nonpending best version record.

When the Suspect status is removed from the record, the result is sent to those applications that do not yet have the record. By default two system preferences allow you to use suspect match:

- **Enable CDM SM 1.** This value enables suspect match on account records.
- **Enable CDM SM 2.** This value enables suspect match on contact records.

Scenario for Creating a New Record with Suspect Matching

This topic gives the following example of how Oracle Customer Hub (UCM) might be used for suspect matching. You might use Oracle Customer Hub (UCM) differently, depending on your business model. For information on new or enhanced UI objects, see *Suspect Match Object Additions*.

A business user takes an order from a customer at a given address. The order is approved by the system and the user submits the order. Later, when the data steward logs in to the provisioning system to check the delivery date, the new customer record cannot be found. A match is made by address, but the customer name is spelled differently. The business user informs the data steward that two records have been flagged as potential matches by Oracle Customer Hub (UCM) and are waiting for her to review them. After verifying the data, the data steward merges the two records into one master record.

Later the user notices that the record in the call center application shows several previous orders in addition to the one just taken, and she notices that one of the orders has a status of In Process. If she were to combine the order, her customer would qualify for free shipping, so she combines the two orders, then sends the customer an email informing him of the new combined order and the refund for the previously charged shipping cost.

Configuring Suspect Match

Suspect match might be used with contact and account records. In previous releases of Oracle Customer Hub (UCM), the Incoming Duplicates (Link and Update) views were show by default. To facilitate suspect match, the Incoming Duplicate view must be hidden. Use the following task to configure suspect match.

To configure suspect match

1. Navigate to Administration - Application, then Views.
2. Query for the following views:
    - UCM Account DeDup View
    - UCM Contact DeDup View
3. Delete both records.
4. Navigate to Administration - System Preferences.

   The System Preferences screen appears.
5. In the System Preference Name field of the System Preferences screen perform the following query:

   ```
   Enable CDM*
   ```
6. Make sure the following values are present:
    - **Enable CDM SM 1.** This value enables Suspect Match on account records.
    - **Enable CDM SM 2.** This value enables Suspect Match on contact records.

Note: If you want to disable Suspect match and enable Link and Update for manual review, see *Enabling the Link and Update Feature*.

Suspect Match Object Additions

This topic includes information about additions applets and applet user properties to enable suspect match. It includes the following topics:

- *New and Enhanced Applets for Suspect Match Object*
- *Suspect Match New Applet User Properties*

New and Enhanced Applets for Suspect Match Object

This topic describes the new and enhanced applets for the suspect match object.

UCM DeDuplication - Account Duplicate Master List Applet

The following information lists the properties of the new list column for the UCM DeDuplication-Account Duplicate Master List applet.

Field	Value
Display Name	Pending Review
Field	Pending Review Status
HTML List Edit	FALSE
HTML Only	FALSE
HTML Type	Field
Read Only	TRUE
Name	Pending Review Status

UCM DeDuplication - Contact Duplicate Master List Applet

The following table lists the properties of the new list column for the UCM DeDuplication-Contact Duplicate Master List Applet.

Field	Value
Display Name	Pending Review
Field	Pending Review Status
HTML List Edit	FALSE
HTML Only	FALSE
HTML Type	Field
Read Only	TRUE
Name	Pending Review Status

UCM Account Form Applet: New Remove Pending Control

The following table lists the properties of a new control for the UCM Form Applet.

Field	Value
Name	Remove Pending

Field	Value
Caption	Approve
Field	Not applicable
Read Only	Not applicable
HTML Type	Mini button
Method Invoke	RemovePending

UCM Account Form Applet: New Pending Review Status Control

The following table lists the properties of a new control for the UCM Form applet.

Field	Value
Name	Pending Review Status
Caption	Pending Review
Field	Pending Review Status
Read Only	TRUE
HTML Type	Field

UCM DeDuplication - Account Duplicate Master List Applet

The following table lists the properties of the new list column for the UCM DeDuplication - Account Duplicate Master list applet.

Field	Value
Name	Remove Pending
Caption	Approve
Field	FALSE
Read Only	FALSE

Field	Value
HTML Type	Mini button
Method Invoke	RemovePending

Suspect Match New Applet User Properties

This topic describes the new applet user properties for the Suspect Match object.

UCM Account Form Applet: New User Properties

The following information lists the new user properties for the UCM Account Form Applet.

Name	Value
Named Method 1: RemovePending	'INVOKESVC' , 'UCM Data Steward UI Service' , 'RemovePending' , '"Object Type"' , '"Account"' , '"Id"', '[Id]' , '"Pending Status"' , '[Pending Review Status]'
Named Method 2: RemovePending	'INVOKE', 'RefreshBusComp'

UCM Contact Form Applet: New User Properties

The following table lists the new user properties for the UCM Contact Form Applet.

Name	Value
Named Method 1: RemovePending	'INVOKESVC' , 'UCM Data Steward UI Service' , 'RemovePending' , '"Object Type"' , '"Contact"' , '"Id"', '[Id]' , '"Pending Status"' , '[Pending Review Status]'
Named Method 2: RemovePending	'INVOKE', 'RefreshBusComp'

Enabling the Link and Update Feature

In previous releases of Oracle Customer Hub (UCM), Incoming Duplicate views were shown by default. With suspect match, these Incoming Duplicate views have been disabled by default. Perform the following two tasks to enable the link and update feature:

- *Enabling the Link and Update Feature*
- *Enabling the Link and Update Feature*

Use the following task to disable suspect match and enable link and update functionality.

To disable suspect match and enable link and update

1. Navigate to Administration - System Preferences.

   The System Preferences screen appears.
2. In the System Preference Name field of the System Preferences screen, perform the following query:

   ```
   Enable CDM*
   ```
3. Make sure the following values are present:
   - **Enable CDM SM 1.** This value enables Suspect Match on Account records.
   - **Enable CDM SM 2.** This value enables Suspect Match on Contact records.
4. Delete these two values.

Use the following task to enable the Incoming Duplicates views.

To enable Incoming Duplicates views

1. Navigate to Administration - Application, then Views.
2. Query for the following views:
   - UCM Account DeDup View
   - UCM Contact DeDup View
3. Add both records.

Scenario for Using Account and Contact Merge and Unmerge

This topic gives the following example of how Oracle Customer Hub (UCM) might be used. You might use Oracle Customer Hub (UCM) differently, depending on your business model.

A data steward is responsible for deployed customer information found in Oracle Customer Hub (UCM). After loading several batch jobs of contact or account data successfully from subsidiary entities, the data steward reviews the results of the matching processes and merges the records that are identified as potential duplicate records.

> **Note:** The preconfigured UCM Merge Service supports Account and Contact objects.

For information on new or changed UI objects, see *Additions and Enhancements for Account and Contact Merge and Unmerge Object* .

About Merge and Unmerge

Merge requests might be taken for Contact and Accounts records in the UI and through Web services. You can capture any two records in the UI to perform a merge. You can capture associated information during the merge. The associated information will be available and restored to the victim record after the records are unmerged. After the data steward processes the merge request, the requestors receive a merge result message provided that the requestor is registered with publish and subscribe.

When Oracle Customer Hub (UCM) is the customer master and also the service provider, service requestors can invoke only the Submit Merge Request service. Only a data steward can process the merge request and decide whether it is a valid merge request and process the merge. The following conditions must exist for a valid merge:

- The interface of the service is the external system survivor ID and the victim ID.
- The two records in the merge request have to be in the Oracle Customer Hub (UCM) database. Otherwise, the merge request will be rejected.
- The merge request will be put in the request table, and it waits for the data steward to review and execute the merge operation.
- Only the data steward calls the process merge request service. The process merge request service calls the survivorship service to generate the updated version of the survivor record if the Survivorship Engine is enabled. The survivorship service updates all the foreign key fields that are linked to the victim record to survivor record in the Oracle Customer Hub (UCM) database.
- The UCM submit merge request service provides an asynchronous response of the merge results, but the requestor receives a synchronized response for the request status, such as Pending, Rejected, and so on. After the data steward processes the merge request, the requestors get the merge result only if they registered with the publish and subscribe service.

Supported Child Objects for Merge and Unmerge

You can perform merge and unmerge operations on child objects of the Account and Contact objects. As merge and unmerge operations are carried out on the Account or Contact objects, there are child objects for each object whose respective children records are re-parented accordingly. During a merge operation all the records associated to a merged child object are re-parented to the surviving record. During an unmerge operation a child object's records are considered for being re-parented back to the record that is being unmerged. You can view a list of supported child objects considered during an unmerge operation on the Account and Contact object by viewing the respective user properties of the UCM Merge Service.

The UCM Merge Service uses the Contact_Merge_Support_Child0, and Account_Merge_Support_Child0 user properties to determine which child object values will be considered for merge and unmerge operations for the Contact object and Account object respectively. From Siebel CRM 14.0, the Contact_Merge_Support_Child0 has a newly supported value, SM Contact Social Profile, which stores information about a user’s social media profile. The values for these user properties are show in the following table.

Name	Value
Contact_Merge_Support_ Child0	Personal Address;FINCORP Account;FIN Contact Phone;UCM Contact Source Data and History;CIF Contact Reference;Alternate Phone;UCM HE Constituent Address;UCM HE Constituent Name;Communication Address;SM Contact Social Profile;
Account_Merge_Support_ Child0	Contact;CUT Address;FINCORP Loan Account;FINCORP Deposit Account;FIN Business Phone;UCM Account Source Data and History;CIF Account Reference

Configuring Merge Request

The merge request uses the business service, UCM Merge Service, and the user property, SubmitMergeRequest_Support_Object. Merge request is enabled by default. The user property value is Account, Contact. When enabled a menu item, Merge Records for Accounts and Contacts, is available for use. If you want to

disable this functionality, remove the corresponding object name from the user property value. For more information, see the following.

Use the following task to view the Merge Request user property.

To view the Merge Request user property

1. Choose Business Service from the Object Explorer, then select Business Service User Prop.

   If Business Service User Prop does not appear, do the following:

   a. From the Siebel Tools View menu, select Options.
   b. Click the Object Explorer tab.
   c. Click Business Service, and when the child properties appear, select the Business Service User Prop check box, then click OK.
2. In the Business Services screen, query for UCM Merge Service.
3. In the Business Service User Props view, query for SubmitMergeRequest_Support_Object.
4. Verify that Account, Contact are present in the Value field.

   Use the following task to disable the Merge Request user property.

To disable the Merge Request user property

- To disable Merge Request functionality for either Account or Contact, delete the corresponding value from the Value field. Deploy your changes to Siebel runtime Repository.

> **Note:** Completed merge requests can be purged using the Purge button in the UCM Account Merge Request Detail view. The Purge button can also be configured to purge Merge Requests with a status other than Completed. You might configure the Purge button by using the UCM Merge Service business service and the user property, PurgeType. The preconfigured value is set to Complete.

About Guided Merge for Oracle Customer Hub (UCM)

Business Service Name	User Property Name	User Property Default Value
UCM Data Decay Service	Max Decay Indicator	100. Optional. May be customized.
	Notification Action Type	Alert. Optional. May be customized.
	Notification Decay Indicator	80. Optional. May be customized.
UCM Data Steward UI Service	IOForPubSubAccount	SwiOrganizationPublishIO
	IOForPubSubContact	SwiPersonPublishIO

Business Service Name	User Property Name	User Property Default Value
UCM Merge Service	Account/CUT Address	UCM Account CUT Address;Account Id;Address Id
	Account_Merge_Support_Child0	CUT Address;FINCORP Loan Account;FINCORP Deposit Account;FIN Business Phone;UCM Account Source Data and History;CIF Account Reference
	Contact/Personal Address (w/ Primary)	UCM Contact Personal Address;Contact Id;Address Id
	Contact_Merge_Support_Child0	Personal Address;FINCORP Account;FIN Contact Phone;UCM Contact Source Data and History;CIF Contact Reference
	MERGE_OPERATION_TYPE	Merge
	MERGE_PUBLISH_IO_Account	SwiOrganizationPublishIO
	MERGE_PUBLISH_IO_Contact	SwiPersonPublishIO
	PUBLISH_METHOD_TYPE	PublishMethod
	PurgeType	Completed
	SURVIVOR_OPERATION_TYPE	Upsert
	SubmitMergeRequest_Support_Object	Account,Contact
	VICTIM_OPERATION_TYPE	Delete
UCM Survivorship Engine	BRCurrencyFldConstruct_Contact	Income:Income Currency Code,Income Exchange Date
	DefaultAttrGrpName_Account	Default Account Attribute Group
	DefaultAttrGrpName_Contact	Default Contact Attribute Group
	DefaultAttrGrpName_Household	Default Household Attribute Group
	ExemptFields_Account	Party Uld;Id
	ExemptFields_Contact	Party Uld;Person Uld;Id
	ExemptFields_Household	Party Uld;Id

Business Service Name	User Property Name	User Property Default Value
	ExternalBSIOInterface_Account	UCMAccountAttributeGroupMetaData
	ExternalBSIOInterface_Contact	UCMContactAttributeGroupMetaData
	ExternalBSIOInterface_Household	UCMHouseholdAttributeGroupMetaData
	KnowledgeBaseName	ucmsurvivorshiprules
	NULLValueOverride_Account	TRUE
	NULLValueOverride_Contact	TRUE
	NULLValueOverride_Household	TRUE
	CompareDateField_Account	Hidden
	CompareDateField_Contact	Hidden
	CompareDateField_Household	Hidden

Oracle Customer Hub (UCM) Guided Merge provides a user interface that lets you review each step during the merge process and change the final version of a record before the merge operation occurs. You can review duplicate records in several versions, then decide which record to keep and how it appears before saving records to the merge task.

Guided Merge allows you to review duplicate records and proposed merges by presenting three versions of the duplicate records and allowing you to decide how the record will be presented in Oracle Customer Hub (UCM) after the merge task is approved and saved.

Note: Guided Merge displays only parent object values. You cannot intervene in the child merge process.

Four versions of a merged record are as follows:

- **Victim.** Indicates the record that will be deleted from the master business component.
- **Survivor.** Indicates the record that will be saved to the master business component.
- **Suggested.** By default, indicates the survivor record values if survivorship is not enabled for the object in consideration.
- **Final version.** Indicates the final version of the record based on the user's decision.

Scenario for Using Guided Merge

This topic gives the following example of how Oracle Customer Hub (UCM) might be used. You might use Oracle Customer Hub (UCM) differently, depending on your business model.

A data steward must merge two contact records. Record A contains First Name, Last Name, and Email ID. Record B contains First Name, Last Name, and Fax number. Record A is considered the surviving record, and Record B is considered the victim record. The data steward merges these two records and decides to require First Name, Last Name, Email ID, Phone Number and Mobile Phone Number. The data is entered in the columns in the UI, and the data steward submits a merge request. The resultant surviving record contains the following: First Name, Last Name, Email ID, Phone Number, and Mobile Phone Number.

Configuring Guided Merge

To enable Merge and Guided Merge, you must activate the process properties for the Process Merge Request Workflows, enable the Process Merge Request Workflows themselves, then activate Merge tasks. Use the following task to activate process properties for Process Merge workflows.

To activate process properties for Process Merge Request workflows

1. From Object Explorer, click Workflow Process, the WF Process Prop.
2. In the Workflow Processes list, Query for UCM Process Merge Request in the Process Name field.
3. In the WF Process Props list, find the following:
    - EnablePubSub
    - EnableSE
4. Change the values for both default strings from False to True.

    Note: Change these values only if you want to enable Publish and Subscribe or the Survivorship Engine.
5. Deploy and Activate the workflow.

Use the following task to enable Process Merge Request workflows.

To enable Process Merge Request workflows

In Web Tools or Siebel Tools query for the UCM Process Merge Request Workflow Process. For more information on Siebel workflows and how to activate them, see *Siebel Business Process Framework: Workflow Guide*

Use the following task to enable merge request tasks.

To enable merge tasks

1. Make sure the following Tasks are active. For more information on Siebel workflows and how to activate them, see *Siebel Business Process Framework: Workflow Guide*
    - UCM Merge Account Request Task
    - UCM Merge Account Task

- UCM Merge Contact Request Task
- UCM Merge Contact Task

Use the following task to add additional fields to the Guided Merge user interface.

To add additional fields to the Guided Merge user interface

1. From the Object Explorer, select Integration Object, then query for either UCMAccountTaskUIMerge or UCMContactTaskUIMerge.
2. In the Object Explorer, click Integration Component, then in the Integration Components view select either Account or Contact.
3. Right-click and select New Record, then add the Integration Component field that you want to make available in the Guided Merge Task UI.

   Note: Make sure the External Name property of the Integration Component Field maps to the correct Business Component field.

4. Validate the integration object, and deploy changes to Siebel Runtime Repository.
5. Restart Oracle Customer Hub (UCM).

Additions and Enhancements for Account and Contact Merge and Unmerge Object

This topic describes the additions or enhancements to the Account and Contact Merge and Unmerge objects. It includes the following:

- *New and Enhanced Views for the Merge and Unmerge Object*
- *New and Enhanced Applets for the Merge and Unmerge Object*

New and Enhanced Views for the Merge and Unmerge Object

This topic describes the new and enhanced views for the Merge and Unmerge object.

UCM Account Merge Request Detail View

The following table displays the additions or enhancements to the UCM Account Merge Request Detail view.

Business Object	View Web Template Items
UCM Merge Request	UCM Account Merge Request List Applet
	UCM Merge Account Entry Applet
	UCM Victim Account List Applet

UCM Contact Merge Request Detail View

The following table displays the additions or enhancements to the UCM Contact Merge Request Detail view.

Business Object	View Web Template Items
UCM Merge Request	UCM Contact Merge Request List Applet
	UCM Merge Contact Form Applet
	UCM Victim Contact List Applet

New and Enhanced Applets for the Merge and Unmerge Object

This topic describes the new and enhanced applets for the Merge and Unmerge object.

UCM Merge and Unmerge Applets

The following table displays the new applets associated with UCM Merge and Unmerge functionality.

New Applet	Business Component
UCM Contact Merge Request List Applet	UCM Merge Request
UCM Account Merge Request List Applet	UCM Merge Request
UCM Victim Account List Applet	Victim Account
UCM Victim Contact List Applet	Victim Contact
UCM Account List Applet - Unmerge	Copy of the SIS Account List Applet. The search specification was added to limit the view only to accounts associated with the merge event.
UCM Contact List Applet - Unmerge	Copy of Contact List Applet. The search specification was added to limit the view only to contacts associated with the merge event.
UCM Merge Contact Form Applet	Copy of Contact Form Applet. Read only.
UCM Merge Account Entry Applet	Copy of SIS Account Entry Applet. Read only.

New Control for Launch Guided Merge

The following table lists the applets that contain this control.

Control	Applet
Launch Guided Merge	UCM Account Duplicate List Applet. It is in the UCM Account Duplicates Detail view.
	UCM Contact Duplicate List Applet. It is in the UCM Contact Duplicates Detail view.

Control	Applet
	UCM Victim Account List Applet. It is in the UCM Account Merge Request Detail view
	UCM Victim Contact List Applet. It is in the UCM Contact Merge Request Detail view

Administering Duplication Resolution Functionality

To administer the duplicate-resolution functionality in Oracle Customer Hub (UCM), complete the following tasks:

- *Managing Duplicate Records from External Systems with Suspect Match*
- *Managing Duplicate Records from External Systems with Link and Update*
- *About Merge and Unmerge*
- *Merging Duplicate Records in Oracle Customer Hub(UCM)*
- *Unmerging Previously Merged Records in Oracle Customer Hub (UCM)*

Managing Duplicate Records from External Systems with Suspect Match

For information on managing duplicate records with Suspect Match, see *About Suspect Matching*.

Managing Duplicate Records from External Systems with Link and Update

Duplicate records in Oracle Customer Hub (UCM) are identified and listed in the Incoming Duplicates screen. This screen displays incoming record updates from external systems that match one existing record within Oracle Customer Hub (UCM) below the automatic update threshold or match to multiple existing records within Oracle Customer Hub (UCM) above the manual update threshold. Administrators can evaluate the records and decide whether to update through one of two ways: Link and Update, or Suspect Match.

When updating records by selecting the Link and Update button, Oracle Customer Hub (UCM) does the following:

- 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 record in Oracle Customer Hub (UCM)
- Calls the survivorship rules to produce a new best version of the record

For information on other Oracle Customer Hub (UCM) data management tasks, see *Configuring Source Data History and Cross-Reference Functionality*

Use the following task to update or promote duplicate records.

To update or promote duplicate Oracle Customer Hub (UCM) records

1. Navigate to Administration - Universal Customer Master, then Incoming Duplicates.
2. Click either the Duplicate Accounts View tab or Duplicate Contacts View tab.
3. In the Account Source or Contact Source list, select the source record of interest.

   The records in this list are identified as duplicates of the base Oracle Customer Hub (UCM) record in the lower view. The record columns provide details on each account or contact, including the external system where the record originated.
4. In the lower Duplicate tab, review the Oracle Customer Hub (UCM) record that is in conflict with the records in the Source list.

   The record columns in this list provide details on the existing account or contact in Oracle Customer Hub (UCM). The score field determines how close the incoming record matches the existing record.
5. 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.

About Merge and Unmerge

Merge requests can be made for Contact and Account records in the Oracle Customer Hub (UCM) user interface, and through Web services. You can capture any two records in the user interface to perform a merge. You can capture associated information during the merge. The associated information will be available and restored to the victim record after the records are unmerged. After the data steward processes the merge request, the requestors receive a merge result message providing that the requestor is registered with the publish and subscribe service.

When Oracle Customer Hub (UCM) is the customer master application and also the service provider, service requestors can invoke only the Submit Merge Request service. Only a data steward can process the merge request and decide whether it is a valid merge request and process the merge. The following conditions must exist:

- The interface of the service must contain the external system survivor ID and the victim ID.
- The two records in the merge request must be in the Oracle Customer Hub (UCM) database. Otherwise, the merge request will be rejected.
- The Merge Request service will be put in the request table and waits for the data steward to review and execute the merge operation.
- The Process Merge Request service will be invoked by the data steward only. It calls the Survivorship service to generate an updated version of the survivor record if the Survivorship Engine is enabled. It updates all foreign key fields linked to the victim record to survivor record in the Oracle Customer Hub (UCM) database.
- UCM Submit Merge Request service provides an asynchronous response of the merge results, but the requestor receives a synchronized response for the request status, such as Pending, Rejected and so on. After the data steward processes the merge request, the requestors will get the merge result only if they registered with the publish and subscribe service.

Merging Duplicate Records in Oracle Customer Hub (UCM)

Oracle Customer Hub (UCM) identifies and lists duplicate contact and account records for review by administrators after running data quality and 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 record in Oracle Customer Hub (UCM). The merging process does the following:
  - Stores the duplicate record as the 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 record unchanged in Oracle Customer Hub (UCM).

Use the following task to review and manage duplicate records.

To review and manage duplicate Oracle Customer Hub (UCM) records

1. Navigate to Administration - Universal Customer Master, then Existing Duplicates.
2. Select the Duplicate Contacts or Duplicate Accounts tab.

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

   The Duplicate Account Resolution or Duplicate Contact Resolution view appears. The first record is the current Oracle Customer Hub (UCM) version of the account or contact. There is a duplicate record at the end of the view.
4. Resolve the duplicate record by:
   a. Merging the duplicate record with the Customer Hub record, by clicking the Merge button.

      Note: Before merging a duplicate record, Oracle recommends generating the hierarchy. To generate the hierarchy choose navigate to Accounts, and then in the Global Accounts Administration screen click the Generate Hierarchy button.

   b. Deleting the duplicate record by clicking the Menu button and then Delete Record.
   c. Deleting the Oracle Customer Hub (UCM) record by clicking the Delete button.

Duplicate Resolution of Suspects Within Existing Duplicates View in B2C Deployment

The UCM Contact Duplicates Detail view is preconfigured for business-to-consumer (B2C) deployment. The addresses of the accounts associated with a contact are shown in this view.

Note: You access the UCM Contact Duplicates Detail view by drilling down on the last name in the Contact Duplicates view under the Existing Duplicates screen.

For B2C deployments complete the following configuration task to enable contact addresses to be displayed in the UCM Contact Duplicates Detail view.

To enable contact addresses in the UCM Contact Duplicates Detail view

1. In the Object Explorer, click Business Component.
2. Query for the following business component: UCM DeDuplication Results (Contact).
3. Query for the field, Primary Address ID.
4. Change the Join property from S_ORG_EXT to S_CONTACT, and change the Property column from PR_ADDR_ID to PR_PER_ADDR_ID.
5. Deploy changes to Siebel Runtime Repository.

Unmerging Previously Merged Records in Oracle Customer Hub (UCM)

An Oracle Customer Hub (UCM) record that results from the merging of two previously merged records, as defined by the survivorship rules, can be unmerged to their original state. After unmerging the merged record, Oracle Customer Hub (UCM) restores the parent victim record and reverts the parent merged record back to its previous state. Oracle Customer Hub (UCM) removes the children of the victim record from the surviving merged record and adds them back to the victim record.

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

To unmerge a merged Oracle Customer Hub (UCM) record

1. Navigate to Administration - Universal Customer Master, then UnMerge Profiles.
2. Click either the Merged Accounts View tab or Merged Contacts View tab.

   A list of merged records is displayed.
3. In the Accounts or Contacts, select the source record of interest.
4. Click the Unmerge button.

7 Configuring the Survivorship Settings for Oracle Customer Hub (UCM)

Configuring the Survivorship Settings for Oracle Customer Hub (UCM)

This chapter describes how to configure the survivorship settings for Oracle Customer Hub (UCM). It includes the following topics:

- *Configuring Survivorship Settings for Oracle Customer Hub (UCM)*
- *Disabling Data Quality Synchronization in UCM Transaction Manager Business Service*
- *Configuring Survivorship to Use An External Rules Engine*
- *Process of Creating Survivorship Rules*
- *About Default Survivorship Rules in Oracle Customer Hub (UCM)*
- *Example Process of Creating a New Attribute Group for the Contacts Default Rule*
- *New and Enhanced Applets for the Survivorship Engine*

Configuring Survivorship Settings for Oracle Customer Hub (UCM)

The following topics detail configuration tasks for survivorship for Oracle Customer Hub (UCM):

- *About Oracle Customer Hub (UCM) Survivorship Settings*
- *Configuring UCM Transaction Manager for Data Management and Survivorship*
- *New and Enhanced Applets for the Survivorship Engine*

About Oracle Customer Hub (UCM) Survivorship Settings

Survivorship is the process of creating a best version record, known as a surviving record. When multiple systems attempt to contribute information about a customer, survivorship rules are used to govern how to construct a best version record, the surviving record, for the specific data object. Each rule is applicable to only one data object, and within each rule, multiple attribute group sets can be associated. Survivorship algorithms enable you to have greater control over the survivorship rules applied to the update and merge transactions. You can include conditions, such as Completion of Group, which designates the most complete combination of the Name and Main Phone Number fields is the retained Account record.

From Siebel CRM, Version 8.1.1.9 on, you can define survivorship rules for any child entity of the Account, Contact or Household object. You define survivorship rules for child entities in the UCM Entity Rule Set View.

From Siebel CRM 14.0, you can define survivorship rules for Contact Social Media Profile as child entity of the Contact object.

Additionally, you might use the Golden Wins and Contributor Wins survivorship rules. The Golden Wins rule stipulates that the values stored in Oracle Customer Hub (UCM) always win over the incoming record, or in the case of a merge operation, the surviving record's value always wins over the victim's value. The Contributor Wins rule stipulates that the values sent by an external application to Oracle Customer Hub (UCM) always win over the values in the existing best version record, or in the case of a merge operation, the victim record's values will be retained over the surviving record's value.

Survivorship rules govern how the best version record can be constructed for the specific data object. Each rule is applicable to only one data object. Within each rule, though, there can be multiple attribute group sets associated with the rule.

Attribute group rule intersection objects are in specific survivorship rules. These objects are used to govern how the specific attribute group functions within the given rule when evaluated by the UCM Survivorship Engine. For information on configuring UCM Survivorship Engine properties, see *Configuring UCM Transaction Manager for Data Management and Survivorship*.

Definitions of Survivorship Terms

This topic describes terms used in the implementation of survivorship.

- Cross-Reference. Tables within Oracle Customer Hub (UCM) to link the Unique Id (UID) within Oracle Customer Hub (UCM) to the Source Id from each subscribed application.
- Data Consolidation. The process of consolidating incoming records contributed by source systems with matching best version records within Oracle Customer Hub (UCM), while applying survivorship to govern the update of fields.
- Merge. The process of combining two or more duplicate best version records into a single surviving record. During the merge process, survivorship is applied, and appropriate audit trail information is recorded.
- Data Recovery. The process of reverting to historical versions of the best version record from an erroneous update or merge operation.
- Victim Record. The record that is merged into another record during the merge process. This record is usually not retrievable after a merge.
- Surviving Record. The record into which another record or records are merged during a merge process. This record remains in Oracle Customer Hub (UCM) after a merge.
- Source Confidence Level. The level of confidence on the validity of given attribute group field values from a specific source. Even though the source confidence level is specified in numbers, the importance is the relative value among sources, not the absolute value.

Configuring UCM Transaction Manager for Data Management and Survivorship

User Property	Default Value
DefaultAttrGrpName_Account	Default Account Attribute Group
DefaultAttrGrpName_Contact	Default Contact Attribute Group

User Property	Default Value
ExemptFields_Account	Party Uld; Id
ExemptFields_Contact	Party Uld; Person Uld; Id

Management and Survivorship

You can enable customer data management of the UCM Survivorship Engine for the UCM Transaction Manager business service by configuring its user properties. For information on configuring business service user properties using Siebel Tools, see *Integration Platform Technologies: Siebel Enterprise Application Integration* . For background information on the UCM Transaction Manager business service, see *UCM Transaction Manager*.

To configure UCM Transaction Manager to enable data management and the UCM Survivorship Engine

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

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

Disabling Data Quality Synchronization in UCM Transaction Manager Business Service

> **Note:** This topic is applicable only if you are using Oracle Data Quality as your Matching Engine. Disable this process property with caution, as doing so also disables data quality synchronization between Oracle Customer Hub (UCM) and Oracle Data Quality Matching Server.

Oracle Customer Hub (UCM) features both coarse grain and fine grain Web services for all key entities such as Person, Organization, Group, and Financial Account. All requests that come into Oracle Customer Hub (UCM) from external systems come by way of coarse grain Web services, and each request goes through the UCM Transaction Manager and the IIR server. A set of mandatory fields must be passed to the Siebel integration object whenever a synchronization operation with the IIR server is called from the UCM Transaction Manager.

Synchronization can reduce performance if you have, for instance, multiple address deduplication configured. While CRUD operations do not tax system resources in the same way, there is the potential for performance issues when executing operations such as updating party relationships, or updating specific non matching fields, such as Secondary Email Address, Fax Number, and so on.

A process property called Turn On DQ External Sync for the workflow which uses the UCM Transaction Manager business service allows you to disable external data synchronization. The process property functions as a switch and is enabled by default. If you set the process property to False, a synchronization call from the UCM Transaction Manager business service to the IIR server is not made.

To disable data quality synchronization for the UCM Transaction Manager business service

1. In Siebel Tools, lock the relevant project.
2. In the Object Explorer, select Workflow Process, then query for YYY in the Object List Editor.
3. Select the YYY workflow in the Workflow Processes list.
4. In the Object Explorer, select WF Process Prop, then add Turn On DQ External Sync in the Object List Editor.
5. Add the value of Turn On DQ External Sync to False to disable data quality synchronization.

   Turn On DQ External Sync is set to True by default.

Configuring Advanced Survivorship Features

To configure the advanced survivorship features perform the following tasks:

- *Configuring Survivorship Custom Date Fields*
- *Configuring Oracle UCM to Send User Keys for Child Records to xternal Rules Engine*
- *Configuring Survivorship to Use An External Rules Engine*

Configuring Survivorship Custom Date Fields

By default, the survivorship feature uses the Updated field to do comparison rules, such as Recent and History. You can configure any date or time field to use with the survivorship feature. For example, if you want to use the Contact birthday field as your comparison timestamp, you add the business service user property, CompareDateField_Contact, with the value, Birth Date. After these changes, make sure you compile and deploy your changes to Siebel Runtime Repository and restart the Siebel Server.

Configuring Oracle UCM to Send User Keys for Child Records to xternal Rules Engine

If you use an external rules engine for rules evaluation, then you can optionally configure Oracle Customer Hub (UCM) to include user key information in each of the child records that are sent to the rules engine. Providing this information can help the rules engine to match child records.

The user key information is sent in the <UserKey> property in the child records. Because Oracle Customer Hub (UCM) can't determine which user key an external rules engine uses, all of the user keys are sent. The format of the <UserKey> property is as follows:

```
<UserKey>UserKey1:FieldValue1FieldValue2…FieldValuen;UserKey2:FieldValue1FieldValue2…FieldValuen;…
UserKeyn:FieldValue1FieldValue2…FieldValuen</UserKey>
```

where:

- FieldValue1 is the value in the first field in the user key, FieldValue2 is the value in the second field in the user key, and so on.
- FieldValuen is the value in the last field in the user key.
- UserKeyn is the last user key for the record.

As an example, the user keys for the Alternate Phone object are as follows:

- UserKey1: Alternate Phone Id
- UserKey2: Alternate Phone Integration Id
- UserKey3: Alternate Phone # + Alternate Phone Name

> **Note:** Alternate Phone Name is Address + Name.

If you configure Oracle Customer Hub (UCM) to include user key information in each of the child records, then the following is an example of the output for a contact record that has three alternate phone child records.

```
<ListOfContact>
 <Contact>
 <Id>88-1VW1ZN</Id>
 <FirstName>Jane</FirstName>
 <LastName>Lee</LastName>
 <EmailAddress>jane.lee@exampleco.com</EmailAddress>
 <WorkPhone>6035550185</WorkPhone>
 <ListOfAlternatePhone>
 <AlternatePhone>
 <UserKey>UserKey1:88-1VW3UE;UserKey2:;UserKey3:3333333333From SAP01</UserKey>
 <UseType>Home</UseType>
 <Id>88-1VW3UE</Id>
 <Address>3333333333</Address>
 <Name>From SAP01</Name>
 </AlternatePhone>
 <AlternatePhone>
 <UserKey>UserKey1:;UserKey2:;UserKey3:2222222222From Siebel01</UserKey>
 <UseType>Personal</UseType>
 <Id>Dummy1</Id>
 <Address>2222222222</Address>
 <Name>From Siebel01</Name>
 </AlternatePhone>
 <AlternatePhone>
 <UserKey>UserKey1:88-1VW3UF;UserKey2:;UserKey3:1111111111From Siebel02</UserKey>
 <UseType>Business</UseType>
 <Id>88-1VW3UF</Id>
 <Address>1111111111</Address>
 <Name>From Siebel02</Name>
 </AlternatePhone>
 </ListOfAlternatePhone>
 </Contact>
</ListOfContact>
```

The following procedure describes how to configure Oracle Customer Hub (UCM) to send user key information in child records.

To configure Oracle Customer Hub (UCM) to send user keys for child records to an external rules engine

- In the UCM Survivorship Engine business service, add a new user property, Enable Sending User Key Field to External Business Service, with the value set to True.

Configuring Survivorship to Use An External Rules Engine

You can configure survivorship to use an external rules engine for rules evaluation.

To configure survivorship to use an external rules engine

1. Navigate to Administration - Universal Customer Master, then Survivorship Rules.

   This view enables you to configure the rules engine for survivorship evaluation at two levels: default criteria or attribute group comparison rules.
2. In the Default Criteria field, select External.
3. In the Business Service Name field, choose the business service name for the external rules engine from the list.
4. In the Business Service Method field, choose the business service method name for the external rules engine.

Administering Survivorship for Oracle Customer Hub (UCM)

Parameter	Value
DeDuplication Data Type	EDQ or ISS
UCM Data Management Match Flag	TRUE

System Preference	Value
Enable CDM EM 1	Account
Enable CDM EM 2	Contact
Enable CDM Match 1	Account
Enable CDM Match 2	Contact

Parameter	Value
DeDuplication Data Type	EDQ or ISS

Parameter	Value
DeDuplication Enable Flag	TRUE

Name	Value
Enable Deduplication	Yes
Key Type	Standard
Match Threshold	90
Search Type	Typical

The following topics detail survivorship administration tasks for Oracle Customer Hub (UCM):

- *About Oracle Customer Hub (UCM) Survivorship Rules*
- *Example of Using Survivorship Rules*
- *Process of Creating Survivorship Rules*
- *Creating Attribute Groups for Survivorship Rules*
- *Creating Oracle Customer Hub (UCM) Survivorship Rules*
- *Activating Oracle Customer Hub (UCM) Survivorship Rules*
- *Example Process of Creating a New Attribute Group for the Contacts Default Rule*
- *Example Process of Creating a New Attribute Group for the Contacts Default Rule*
- *Attaching the New Attribute Group to the Contact Default Rule*

About Oracle Customer Hub (UCM) Survivorship Rules

Survivorship rules are an automated means of doing the following:

- Controlling the quality of customer data stored in Oracle Customer Hub (UCM)
  Multiple systems connect to Oracle Customer Hub (UCM) with record update and insert requests.
- Making sure that Oracle Customer Hub (UCM) presents the most trusted information contributed by each system

Administrators can create and use survivorship rules to govern which updates from which external systems can be trusted at the field level based on the key criteria. Survivorship rules use attribute groups that determine the set of fields to be evaluated by the survivorship rule. The rules use comparison criteria, such as the 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 Oracle Customer Hub (UCM).

There is an implicit default attribute group which includes the fields that are not explicitly defined in an attribute group definition. Each survivorship rule has a default criterion. If there is no explicit rule definition for a certain attribute group

(including the default attribute group), then the default criterion applies to determine whether an external system can update Oracle Customer Hub (UCM).

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

About Oracle Customer Hub (UCM) Child Survivorship

Child survivorship allows administrators to set survivorship rules for child entities of records, such as personal address, financial account, email, phone and so on. Administrators can also set survivorship rules for custom child entities.

Comparison Methods Used by the Survivorship Rules

The survivorship rules use the following comparison methods to evaluate the quality of the object fields associated with an attribute group:

- **Recent.** Compares the date for the field data of the object attribute group against the date of the updating message. The most recent data survives.
- **History.** Compares the date for the field data of the object attribute group against the date of the updating message. The oldest data survives.
- **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 with the highest confidence level survives.
- **External.** Indicates the attribute group settings that enable you to configure any supported rules engine to set specific survivorship rules.
- **Golden.** The Golden Wins rule stipulates that the values stored in Oracle Customer Hub (UCM) will always win over the incoming record, or in the case of merge, the surviving record's value will always win over the victim's.
- **Contributor.** The Contributor Wins rule stipulates that the values sent by external application into Oracle Customer Hub (UCM) will always win over the values in the existing best version record, or in the case of merge, the victim record's values will be retained over the surviving record's.

Restrictions When Defining the Survivorship Rules

The following restrictions and recommendations must be considered when defining survivorship rules:

- Use only one active rule for each object, although administrators can define multiple rules for each object (account, contact).
- Include each individual field within a given object in one attribute group definition. Otherwise, conflicts might occur when the survivorship rules are evaluated.

Note: You create survivorship rules for primary objects in the UCM Rule Set Attribute Group view, and use the UCM Entity Rule Set view to create survivorship rules for child objects.

For information on the process of creating survivorship rules, see *Process of Creating Survivorship Rules*.

Example of Using Survivorship Rules

This topic gives one example of how to access and use Oracle Customer Hub (UCM). You might use the process differently, depending on your business requirements.

Two records with similar data are entered. The first record is entered with the First Name, Last Name and birth date. The other record is entered with the First Name, Last Name, Birth Date and SSN number. Based on the rule in the external rules engine, the surviving record will display with the completeness of the record.

Example of Using Child Survivorship Rules

This topic gives one example of how to access and use Oracle Customer Hub (UCM). You might use the process differently, depending on your business requirements.

By default the Contact Personal Address object has a business rule defined as follows: Address Type is defined as Business, and Comparison Rule is defined as Golden. So, if an incoming request and an existing matching record each have the same business type but have personal address fields which are different, the personal address record of the existing contact wins.

Considerations When Using Child Survivorship Rules

This example demonstrates Source and Recent Rule of Entity Survivorship with the Personal Address Child object.

Rules of Execution

- During an Insert operation Entity Survivorship rules are not applied, but attribute data is captured.
- The default criteria for user key-based matches are used to eliminate the losing record in case of a user key-based match.
- Internal Resolution between child records is performed either within the existing master record or with the incoming match record using the default criteria.
- When using Field Value Based rules, the rule executed on the child record whose field value matches those defined, the record with the lowest sequence number has the highest rank.
- If you are using the Same Field Value Based rule, and the Comparison Field Value provided is empty then the Entity Survivorship engine looks for an Exact Value match child record to apply the comparison rule. This is typically used on FINCORP Loan, or deposit account objects to apply the rule to obtain an exact account number.

Setup

Use the Personal Address child object of the Contact parent object with the Default Criteria set to Recent. The following information lists the comparison rule details.

Sequence Number	Comparison Field Name	Comparison Field Value	Comparison Rule
1	Address Type	<value 1>	Recent
2	Address Type	<value 2>	Source

Note the additional information shown in the following table:

System ID	Source	Value	User Property
Source Confidence	Ebiz01	70	EntityInternalProcess_ PersonalAddress enabled (True).
	Siebel01	80	EntityInternalProcess_ PersonalAddress enabled (True).

Execution Examples

1. Oracle Customer Hub (UCM) receives a Contact record (C1) from the system Ebiz01 with following three Personal Addresses:
    - Personal Address Record A1: INSPersonalAddressType = Value 1
    - Personal Address Record A2: INSPersonalAddressType = Value 2
    - Personal Address Record A3: INSPersonalAddressType = Value 2

    The result is that attribute data from the Contact record is inserted with three personal address records.
2. Oracle Customer Hub (UCM) receives a Contact record (C2) from the system Siebel01 with three different Personal Addresses:
    - Personal Address Record A4: INSPersonalAddressType = Value 1
    - Personal Address Record A5: INSPersonalAddressType = Value 2
    - Personal Address Record A6: INSPersonalAddressType = Value 2
3. As Contact record C2 is similar to C1 from the match result, the Survivorship engine is invoked to generate the surviving record with resultant children records.

Now the user key based matches are identified and default criteria is applied to eliminate the losing child record. The contact record C1 has the Address A1 attribute and the contact record C2 has the Address attribute A4, and both have the same user keys, and the default criteria of Recent is applied the Address A4 of Contact Record C2, being the most recent record, prevails.

Now two contact records' child records are subjected to field value based rules:

- Since Address Type = Value 1 has already been resolved using the user key based match Contact Record C2's A4 will be part of final surviving record.
- Since Address Type = Value 2 has the Source comparison rule, but before the rule can be applied the child records are run through the Internal Resolution process explained as following:

If EntityInternalProcess_PersonalAddress is set to True, the Contact C1 record undergoes internal resolution for addresses A2 and A3 which have the same INSPersonalAddressType of Value 2. Since the default criteria is set to Recent the most recent of the two versions prevails, in this case, A3. The A2 record's end date column is updated with the current date and time.

Also, if EntityInternalProcess_PersonalAddress is set to True, the Contact C2 record undergoes internal resolution for addresses A5 and A6 which have the same INSPersonalAddressType of Value 2. Since the default criteria is set to Recent the most recent of the two versions prevails, in this case, A6. The A5 record's end date column is updated with current date and time.

- Applying the Field Value based rule (which is set to Source) for INSPersonalAddressType = <Value 2> the source confidence of System = Siebel01(80) is greater than Ebiz01(70) so the A6 record prevails over the A3 record. The A3 record's end date column is updated with the current date and time.

So, the result is that Contact (C12) with five Personal Address records (and A4 and the A1 records with the same user key, INSPersonalAddressType = Value 1) prevailed and the A6 record with INSPersonalAddressType = Value 2 prevailed while the A2, A3 and A5 address records are end dated.

The attribute data is also inserted for new address records and updated for the previously existing address records accordingly.

Process of Creating Survivorship Rules

To create survivorship rules for Oracle Master Data Management Applications, perform the following tasks:

1. *Creating Attribute Groups for Survivorship Rules*
2. *Creating Oracle Customer Hub (UCM) Survivorship Rules*
3. *Creating Oracle Customer Hub (UCM) Survivorship Rules for Child Entities*
4. *Activating Oracle Customer Hub (UCM) Survivorship Rules*

For background information on the survivorship rules, see *About Oracle Customer Hub (UCM) Survivorship Rules*.

Creating Attribute Groups for Survivorship Rules

This task is a step in *Process of Creating Survivorship Rules*. For more information on survivorship rules, see *About Oracle Customer Hub (UCM) Survivorship Rules*. Use the following task to create an attributed group.

To create an attribute group

1. Navigate to Administration - Universal Customer Master, then the Survivorship Rules view.
2. Click the Attribute Group Fields tab.
3. In the Attribute Group list, click the New button.
4. Enter a descriptive name for the attribute group record, and select an object name, for example, Contact, Account, and so on.
5. In the Attribute Group Field Name list, click New.
6. Add one or more fields to define this attribute group.

Creating Oracle Customer Hub (UCM) Survivorship Rules

This task is a step in *Process of Creating Survivorship Rules*. For more information on survivorship rules, see *About Oracle Customer Hub (UCM) Survivorship Rules*. Use the following task to create a survivorship rule.

To create a survivorship rule

1. Navigate to Administration - Universal Customer Master, then Survivorship Rules.
2. In the Rule Name list, click New.
3. Enter values for the columns in the record. See the following table for information on the available columns.

Column Name	Description
Name	Enter a name for the new rule.
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. This column 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.
Start Date	Configures the date when the survivorship rule becomes valid.
End Date	Configures the date when the survivorship rule is invalidated.
Default Rule Set	When no active survivorship rule is defined or the active survivorship rule has expired, the rule with the Default Rule Set that is set to Y is used as the active survivorship rule. Select the Default Rule Set check box for only one survivorship rule for each object.
Business Service Name	This column appears only when the default criteria are set to External. This setting enables you to enter this additional, required information.
Business Service Method	This column appears only when the default criteria are set to External. This setting enables you to enter this additional, required information.

Column Name	Description
Rule Module	This column appears only when the default criteria are set to External. This setting enables you to enter this additional, required information.

4. Click the Attribute Groups tab, and then in the Attribute Groups list, click New to add attribute groups explicitly governed by the survivorship rule.
5. From the Attributes Group window, select the attribute group of focus for this survivorship rule; that is, the attributes that are evaluated by the survivorship rule to determine the most current data.

   You can select preconfigured attribute groups or define unique attribute groups. For information on creating attribute groups, see *Creating Attribute Groups for Survivorship Rules*.
6. Click OK.
7. Enter values for the columns in the record. See the following table for information on the available columns.

Column Name	Description
Attribute Group Name	Select this value from the picklist.
Object Name	Depending on which parent record you are working in, this value will be predefined. For instance, if you are working in a Contact Default record, this object name field will be Contact.
Comparison Rule	Select either External, History, Golden, Recent, Contributor, or Source, as the rule for comparison. Based on the rule selected, Oracle Customer Hub (UCM) evaluates whether the input from the external system can be used to update the specified attribute group values. For more information on each comparison method, see *About Oracle Customer Hub (UCM) Survivorship Rules*.
Sequence	The order in which to evaluate multiple attribute groups.
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 is used as the active survivorship rule. Select the Default Rule Set check box for only one survivorship rule for each object.
Business Service Name	This column with the Business Service Method and Rule Method columns appear only when the default criteria are set to External. This setting enables you to enter this additional, required information.
Business Service Method	This column with the Business Service Name and Rule Method column appear only when the default criteria are set to External. This setting enables you to enter this additional, required information.

8. Step off the record to refresh the view.

9. If the comparison rule for the attribute group is determined by the source, set the source confidence level as follows:
    - Click the Source Confidence tab.
    - In the System ID list, click New.
    - In the new record, define the source system, and set a confidence level for the individual source system.
    - Add other systems as necessary.
10. Add multiple attribute groups, as in the previous step, to further define the survivorship rule.

Creating Oracle Customer Hub (UCM) Survivorship Rules for Child Entities

This task is a step in *Process of Creating Survivorship Rules*. For more information on survivorship rules, see *About Oracle Customer Hub (UCM) Survivorship Rules*. Use the following task to create a survivorship rule for a child record, also known as an entity survivorship rule.

To create a entity survivorship rule

1. Navigate to Administration - Universal Customer Master, then Entity Survivorship Rules.
2. In the Entity Survivorship Rules list, click New.
3. Enter values for the columns in the record. See the following table for information on the available columns.

Column Name	Description
Name	Enter a name for the new rule.
Object Name	Use the drop-down list to choose a valid child record type.
Parent Object	Use the drop-down list to choose a parent object record type.
Status	Sets the entity survivorship rule to Active or Inactive. Select Inactive while creating and defining the entity survivorship rule.
Start Date	Configures the date when the survivorship rule becomes valid.
End Date	Configures the date when the survivorship rule is invalidated.
Default Criteria	The default criteria is used in user key based matches to eliminate the losing record. It is also used within the existing master record or within the incoming match record when the business service user property EntityInternalProcess is set to True.
End Date Field Name	The name of the field for which victim instances will be end-dated.

Column Name	Description
Business Service Name	This column with the Business Service Method and Rule Module column appear only when the default criteria are set to External. This setting enables you to enter this additional, required information.
Business Service Method	This column with the Business Service Name and Rule Method column appear only when the default criteria are set to External. This setting enables you to enter this additional, required information.
Rule Module Name	The Business Rule Modules define in an external rule engine.

Creating Field Value Based Rules for Entity Survivorship

This task is a step in *Process of Creating Survivorship Rules*. For more information on survivorship rules, see *About Oracle Customer Hub (UCM) Survivorship Rules*. Use the following task to create a survivorship rule for a child record, also known as an entity survivorship rule.

To create field value based rules for entity survivorship

1. In Oracle Customer Hub (UCM) navigate to Administration - Universal Customer Master, then click the Entity Survivorship Rules link.
2. In the Entity Survivorship Rules list, click New, then fill in the required information for the entity rule set.
3. In the Comparison Rule applet, click New to enter new Field Value Based rules.
4. Enter the required information for each Field Value Based rule.

   For more information, see *Considerations When Using Child Survivorship Rules*.

   Note: The lower the sequence number you give to the rule, the higher the priority it will be given.

Activating Oracle Customer Hub (UCM) Survivorship Rules

This task is a step in *Process of Creating Survivorship Rules*. For more information on survivorship rules, see *About Oracle Customer Hub (UCM) Survivorship Rules*. Use the following task to activate survivorship rules.

Note: Review Oracle Customer Hub (UCM) survivorship rules thoroughly in a test environment before activating them.

To activate survivorship rules

1. Navigate to Administration - Universal Customer Master, then Survivorship Rules

2. Select the survivorship rule of interest.
3. Click the Summary tab.
4. Review the information contained in this view for the survivorship rule.
5. Click the Attribute Groups tab.
6. Select the survivorship rule of interest, and set the status field to Active.

About Default Survivorship Rules in Oracle Customer Hub (UCM)

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

For more information on survivorship rules, see *About Oracle Customer Hub (UCM) Survivorship Rules*.

Oracle Customer Hub (UCM) includes two default rules named:

- **Contacts Default.** The default criteria field for the Contacts Default rule is set to Recent. This default criteria field 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 Oracle Customer Hub (UCM) record.
- **Accounts Default.** The default criteria field for the Accounts Default rule is history. This default criteria field indicates that the Accounts Default rule evaluates the same incoming updates by date; however, the older data survives in the Oracle Customer Hub (UCM) record.

  There are no defined attribute groups for either of these rules in the attribute group list in the lower view. This setting implies that the default attribute groups are evaluated by the default field in the default survivorship rule.

  The Status field for both rules is set to Active, and the Start Date field for both rules is June 16, 2009, which indicates that this rule is ready to evaluate updates to Oracle Customer Hub (UCM) for the default attribute groups and their associated fields.

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. For more information on survivorship rules, see *About Oracle Customer Hub (UCM) Survivorship Rules*.

Example Process of Creating a New Attribute Group for the Contacts Default Rule

To create a new attribute group for the contacts default rule, do the following:

1. *Creating a New Attribute Group for the Contacts Default Rule*
2. *Attaching the New Attribute Group to the Contact Default Rule*

Creating a New Attribute Group for the Contacts Default Rule

To create a new attribute group for the Contacts Default rule, use the following procedure.Use the following task to create a new rule.

This task is a step in *Example Process of Creating a New Attribute Group for the Contacts Default Rule*

To create a new Attribute Group for the Contacts Default rule

1. Navigate to Administration - Universal Customer Master, then Survivorship Rules.
2. Select the Contact Default rule, then click the Attribute Group Fields tab.
3. In the Attribute Group list, click the New button.
4. Define the new attribute group as follows:

Name	Value
○ **Attribute Group Name**	Income Attribute Group
○ **Object**	Contact
○ **Description**	Defines the rule for the income field in Oracle Customer Hub (UCM)

5. In the Attribute Group Fields list, click the New button.
6. Add the field name Income to the new attribute group from the menu.

   The new attribute group is now defined and references the Income field.

Attaching the New Attribute Group to the Contact Default Rule

To attach a new attribute group for the Contacts Default rule, use the following procedure. Use the following task to attach the newly created rule.

This task is a step in *Example Process of Creating a New Attribute Group for the Contacts Default Rule*

To attach a new attribute group to the contact default rule

1. Navigate to Administration - Universal Customer Master, then Survivorship Rules
2. In the Survivorship Rules list, select the Contact Default rule.
3. Set the Status field to inactive while defining the modification to the default rule.
4. In the Attribute Groups 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 Source.
7. Step off or save the attribute group record for the Contacts Default rule.
8. Because the comparison method is 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 Oracle Customer Hub (UCM) record and replace Source B's data. The final task in the process is to review and activate the enhanced rule. Use the following task to review and activate the newly created 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.

New and Enhanced Applets for the Survivorship Engine

The following topics describe the new and enhanced applets for the survivorship object:

- *New Applet: UCM Entity Rule Set List Applet*
- *New Applet: UCM Entity Rule Set List Applet - External*
- *New Applet: UCM Entity Rule Set List Applet - END Date and External*
- *New Applet: UCM Attribute Field Value Rule List Applet - External*
- *New Applet: UCM Attribute Field Value Rule List Applet*
- *New Applet: UCM Attrib Field Source Confidence Level List Applet*
- *New Applet: UCM Non Source Confidence Level Form Applet*

New Applet: UCM Entity Rule Set List Applet

The following information lists the properties of UCM Entity Rule Set List Applet. The UI controls are: New, Delete, and Query buttons.

Field	Value
Title	Entity Rule Name
Business Component	UCM Entity Rule Set

Field	Value
Class	CSSFrameList
No Delete	False
No Merge	False
No Insert	False
No Update	False

UCM Entity Rule Set List Applet- Web Template

The following table lists the properties of the applet Web template for UCM Entity Rule Set List Applet.

Name	Type	Web Template
Edit	Edit	Applet List Edit (Edit/New/Query)
Base	Base	Applet List (Base/EditList)
Edit List	Edit List	Applet List (Base/EditList)

UCM Entity Rule Set List Applet Columns

The following table lists the properties of the list columns of UCM Entity Rule Set List Applet.

Field	Display Name	HTML Type	Run Time
Default Criteria	Default Criteria	Text	No
Description	Description	Text	Yes
End Date	End Date	Text	Yes
End Date Field Name	End Date Field Name	Text	Yes
Name	Name	Text	Yes
Object Name	Object Name	Text	Yes

Field	Display Name	HTML Type	Run Time
Parent Object Name	Parent Object Name	Text	Yes
Status	Status	Text	Yes
Start Date	Start Date	Text	Yes

New Applet: UCM Entity Rule Set List Applet - External

The following information lists the properties of the UCM Entity Rule Set List Applet - External. The UI controls are New, Delete, and Query buttons.

Field	Value
Title	Entity Rule Name
Business Component	UCM Entity Rule Set
Class	CSSFrameList
No Delete	False
No Merge	False
No Insert	False
No Update	False

UCM Entity Rule Set List Applet - External - Web Template

The following table lists the properties of the applet Web template of UCM Entity Rule Set List Applet - External.

Name	Type	Web Template
Base	Base	Applet List (Base/EditList)
Edit	Edit	Applet List Edit (Edit/New/Query)
Edit List	Edit List	Applet List (Base/EditList)

UCM Entity Rule Set List Applet - External - List Columns

The following tables lists the properties of the list columns of UCM Entity Rule Set List Applet - External.

Field	Display Name	HTML Type	Run Time
Method Name	Business Service Method	Text	Yes
Business Service Name	Business Service Name	Text	No
Default Criteria	Default Criteria	Text	Yes
Description	Description	Text	Yes
End Date	End Date	Text	Yes
End Date Field Name	End Date Field Name	Text	Yes
Name	Name	Text	Yes
Object Name	Object Name	Text	Yes
Parent Object Name	Parent Object Name	Text	Yes
Rule Module Name	Rule Module Name	Text	Yes
Start Date	Start Date	Text	Yes
Status	Status	Text	Yes

New Applet: UCM Entity Rule Set List Applet - END Date and External

The following information lists the properties of UCM Entity Rule Set List Applet - END Date and External. The UI controls are New, Delete, and Query buttons.

Field	Value
Title	Entity Rule Name
Business Component	UCM Entity Rule Set

Field	Value
Class	CSSFrameList
No Delete	False
No Merge	False
No Insert	False
No Update	False

UCM Entity Rule Set List Applet - END Date and External - Web Template

The following table lists the properties of the applet Web template of UCM Entity Rule Set List Applet - END Date and External.

Name	Type	Web Template
Edit	Edit	Applet List Edit (Edit/New/Query)
Base	Base	Applet List (Base/EditList)
Edit List	Edit List	Applet List (Base/EditList)

UCM Entity Rule Set List Applet - END Date and External Columns

The following table lists the properties of the list columns of UCM Entity Rule Set List Applet - END Date and External.

Field	Display Name	HTML Type	Run Time
Method Name	Business Service Method	Text	Yes
Business Service Name	Business Service Name	Text	Yes
Default Criteria	Default Criteria	Text	Yes
Description	Description	Text	Yes
End Date	End Date	Text	Yes
End Date Field Name	End Date Field Name	Text	Yes

Field	Display Name	HTML Type	Run Time
Name	Name	Text	Yes
Object Name	Object Name	Text	Yes
Parent Object Name	Parent Object Name	Text	Yes
Rule Module Name	Rule Module Name	Text	No
Start Date	Start Date	Text	Yes
Status	Status	Text	Yes

New Applet: UCM Attribute Field Value Rule List Applet - External

The following information lists the properties of UCM Attribute Field Value Rule List Applet - External. The UI controls are New, Delete, and Query buttons.

Field	Value
Title	Entity Field Rule Name
Business Component	UCM Attribute Field Value Rule
Class	CSSFrameList
No Delete	False
No Merge	False
No Insert	False
No Update	False

UCM Attribute Field Value Rule List Applet - External Web Template

The following table lists the properties of the applet Web template of UCM Attribute Field Value Rule - External.

Name	Type	Web Template
Edit	Edit	Applet List Edit (Edit/New/Query)
Base	Base	Applet List (Base/EditList)
Edit List	Edit List	Applet List (Edit/New/Query)

UCM Attribute Field Value Rule List Applet - External

The following table lists the properties of the list columns of UCM Attribute Field Value Rule List Applet - External.

Field	Display Name	HTML Type	Run Time
Sequence	Sequence	Text	No
Rule Name	Rule Name	Text	No
Comparison Field Name	Comparison Field Name	Text	No
Comparison Field Value	Comparison Field Value	Text	No
Status	Status	Text	No
Comparison Rule	Comparison Rule	Text	Yes
Business Service Name	Business Service Name	Text	No
Method Name	Method Name	Text	No
Business Service Rule Module Name	Business Service Rule Module Name	Text	No
Description	Description	Text	No

New Applet: UCM Attribute Field Value Rule List Applet

The following information lists the properties of UCM Attribute Field Value Rule List Applet. The UI controls are New, Delete, and Query buttons.

Field	Value
Title	Entity Field Rule Name
Business Component	UCM Attribute Field Value Rule
Class	CSSFrameList
No Delete	False
No Merge	False
No Insert	False
No Update	False

UCM Attribute Field Value Rule List Applet - Web Template

The following table lists the properties of the applet Web template of UCM Attribute Field Value Rule.

Name	Type	Web Template
Edit	Edit	Applet List Edit (Edit/New/Query)
Base	Base	Applet List (Base/EditList)
Edit List	Edit List	Applet List (Edit/New/Query)

UCM Attribute Field Value Rule List Applet Columns

The following table lists the properties of the list columns of UCM Attribute Field Value Rule List Applet.

Field	Display Name	HTML Type	Run Time
Sequence	Sequence	Text	No
Rule Name	Rule Name	Text	No
Comparison Field Name	Comparison Field Name	Text	Yes
Comparison Field Value	Comparison Field Value	Text	Yes
Status	Status	Text	Yes

Field	Display Name	HTML Type	Run Time
Comparison Rule	Comparison Rule	Text	Yes
Description	Description	Text	No

UCM Attribute Field Value Rule List Applet Toggle

The following table lists the properties of UCM Entity Rule Set List Applet Applet Toggle.

Applet	Auto Toggle Field	Auto Toggle Value
UCM Attribute Field Value Rule - External	Comparison Rule INT	External

New Applet: UCM Attrib Field Source Confidence Level List Applet

The following information lists the properties of UCM Attrib Field Source Confidence Level List Applet. The UI controls are New, Delete, and Query buttons.

Field	Value
Title	Attrib Field Source Confidence
Business Component	UCM Attrib Field Source Confidence
Class	CSSFrameList
No Delete	False
No Merge	False
No Insert	False
No Update	False

UCM Attrib Field Source Confidence Level List Applet Web Template

The following table lists the properties of the applet Web template of UCM Attrib Field Source Confidence Level List Applet.

Name	Type	Web Template
Edit	Edit	Applet List Edit (Edit/New/Query)
Base	Base	Applet List (Base/EditList)
Edit List	Edit List	Applet List (Edit/New/Query)

UCM Attrib Field Source Confidence Level List Applet Columns

The following table lists the properties of the list columns of UCM Attrib Field Source Confidence Level List Applet.

Field	Display Name	HTML Type	Run Time
Confidence Level	Confidence Level	Text	No
System Number	System Number	Text	No
System Name	System Id	Text	No
Description	Description	Text	No

UCM Attrib Field Source Confidence Level List Applet Toggle

The following table lists the properties of UCM Attrib Field Source Confidence Level List Applet Toggle.

Applet	Auto Toggle Field	Auto Toggle Value
UCM Non Source Confidence Level Form Applet	Source Rule INT	N

New Applet: UCM Non Source Confidence Level Form Applet

The following information lists the properties of UCM Non Source Confidence Level Form Applet. The UI controls are New, Delete, and Query buttons.

Field	Value
Title	UCM Non Source Confidence Level Form Applet
Business Component	UCM Attrib Field Source Confidence

Field	Value
Class	CSSFrameList
No Delete	False
No Merge	False
No Insert	False
No Update	False

UCM Non Source Confidence Level Form Applet Web Template

The following table lists the properties of the applet Web template of UCM Non Source Confidence Level Form.

Name	Type	Web Template
Edit	Edit	Applet Form Grid Layout
Base	Base	Applet Form Grid Layout

UCM Non Source Confidence Level Form Applet Toggle

The following table lists the properties of UCM Non Source Confidence Level Form Applet Toggle.

Applet	Auto Toggle Field	Auto Toggle Value
UCM Non Source Confidence Level List Applet	Comparison Rule	Source

8 Configuring Publish and Subscribe Functionality

Configuring Publish and Subscribe Functionality

This chapter provides reference information for Oracle Customer Hub (UCM) Publish and Subscribe feature. It includes the following topics:

- *About Publish and Subscribe Functionality*
- *Using Publish and Subscribe for Customized Integration Topologies*
- *Registering Systems Connected to Oracle Customer Hub (UCM)*
- *System Publish and Subscribe Operation Modes*
- *About Outbound PROXY Services*
- *Configuring System Publish and Subscribe for Oracle Customer Hub (UCM)*
- *Modified Table for Oracle Customer Hub (UCM) Version 26.3*

About Publish and Subscribe Functionality

Publish and Subscribe functionality consists of two parts:

1. System Registrations view, an administration UI view, where external systems that want to receive customer data updates from Oracle Customer Hub (UCM) are registered (subscribed). Example of the information provided in this view are the connecting information and preferred message format (e.g., SOAP or JSON) from each external system.
2. UCM Publish/Subscribe Service, a business service that orchestrates and determines the details on how external systems receive customer data updates from Oracle Customer Hub (UCM).

Oracle Customer Hub (UCM) can connect to external systems in two difference modes:

1. **Stand-alone mode**: In this mode, UCM publishes the messages directly to the subscribed systems, using the transport information provided during the system registration using point-to-point integration strategy.
2. **Middleware mode**: In this mode, Oracle Customer Hub (UCM) is connected to subscribed systems via middleware. Oracle Customer Hub (UCM) publishes only one message to middleware with cross reference information of subscribed systems. Cross-reference information is then used by middleware to broadcast messages to the respective subscribed systems.

Starting with release 26.3, Oracle Customer Hub (UCM) provides support for multiple middleware and publishes the messages in hybrid mode in that, UCM can publish the messages to multiple middleware systems, with cross reference information for its own set of subscribed child systems. In addition, UCM publishes the messages directly to each subscribed stand-alone system.

Relevant Features of Hybrid Mode:

1. Each external system can connect to UCM either through a middleware or directly as a stand-alone system.

2. Since this mode supports multiple middleware, each system connecting to UCM through a middleware must specify which parent middleware it belongs to in the System Registration view.
3. All systems connecting to a middleware, will have the same default *Message Type* and transport information specified in parent middleware. The *Message Type* specified in the child system will be ignored.
4. Middleware systems must connect directly to UCM. In other words, middleware systems cannot be daisy chained.

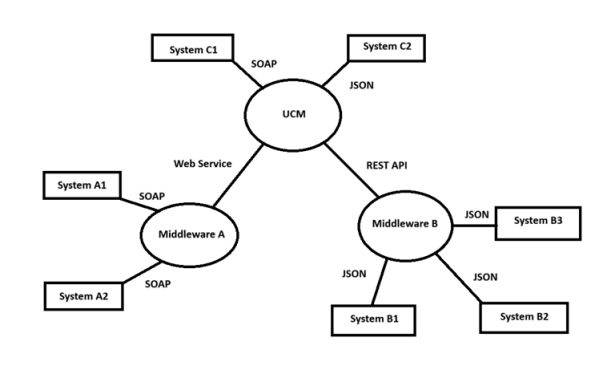
Using Publish and Subscribe for Customized Integration Topologies

Customers can utilize the hybrid mode for configuring a customized integration topology based on their integration requirements. Following are some sample use case scenarios:

Scenario #1: An existing customer connects their systems (applications) with UCM through a middleware using SOAP protocol, since all their systems are using SOAP messaging. A new requirement is to integrate a newer system that uses REST API. Customer has a choice to either connect the new system directly (stand-alone) to UCM using REST protocol or to configure a new middleware using REST protocol and connect the new system to UCM through this new REST middleware.

Scenario #2: A new customer has multiple systems (applications) supporting REST API, and several legacy systems supporting only SOAP message. To integrate all their systems with UCM, customer can configure two middleware systems, one for supporting REST API and the another for supporting SOAP. UCM can be configured to work with these two middleware systems; the external systems can integrate with UCM through the middleware based on the system's supported protocol.

Diagram of a Hybrid Mode Scenario - UCM connects to multiple middleware systems, and multiple stand-alone external systems.



Registering Systems Connected to Oracle Customer Hub (UCM)

Every system that connects to Oracle Customer Hub (UCM) must be registered through the System Registrations view. When the registration is complete, the system's accessibility to Oracle Master Data Management Applications are

administered on the other tabs within this view. For information on further defining the registered system, see the following:

- *Configuring System Publish and Subscribe for Oracle Customer Hub (UCM)*

Use the following task to register a new system with Oracle Customer Hub (UCM).

To register a new system

1. Navigate to the Administration - Universal Customer Master screen, then the System Registrations view.
2. In the System Registrations view, click New.
3. Enter the appropriate information in the available fields to define the system.

   Use the following table for information on each of the fields.

Field	Description
System ID	The System ID represents an identifier for a system that accesses Oracle Master Data Management 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 system that accesses Oracle Master Data Management Applications.
Protocol Type	The protocol that a system uses to access Oracle Customer Hub (UCM). Currently, this value is HTTP, MQSeries, JMS or PROXY.
Middleware System	Checkbox to specify if the system is Middleware or not.
Message Type	Type of message used between the system and Oracle Customer Hub (UCM). The value is JSON or SOAP.
Parent Middleware System	The Parent Middleware System for this system.
Queue Manager Name	The name of the queue manager that receives and sends messages from Oracle Master Data Management Applications. It is applicable only when the protocol type selected is MQSeries.
Queue Receiver Channel	The name of the queue configured on the Queue Manager to receive requests and send responses. It is applicable only when the 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 to the JMS provider. It is applicable only when the protocol type selected is JMS.

Field	Description
Send Queue	Specifies the JMS queue to which messages are sent. Send Queue is applicable only when the protocol type selected is JMS.
Description	The description of the system accessing Oracle Customer Hub (UCM).
Comment	Comments regarding the system accessing Oracle Customer Hub (UCM).

Use the following task to delete a system from Oracle Customer Hub (UCM).

To delete a system

1. Navigate to the Administration - Universal Customer Master screen, then the System Registrations view.
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:** Deleting registered systems can create foreign key integrity violations because external systems might still contain references to deleted foreign keys in operations, such as cross-referencing, survivorship definitions, source data, history records, and so on.
5. Click OK to delete the system.

About System Publish and Subscribe Modes

Oracle Customer Hub (UCM) use one of three modes to publish or update data:

- *Real-Time Mode*
- *Batch Mode*
- *Event Mode*

For information on configuring these modes, see *Configuring System Publish and Subscribe for Oracle Customer Hub (UCM)*.

Real-Time Mode

When configured, the real-time mode of the publish and subscribe functionality runs when another process or business service explicitly calls 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 (Organization, Person, Financial Asset, or Group) Customer Profile Integration SOAP Process, and UCM Batch Process.

Batch Mode

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 the business service 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 for the business component. This step is 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 indicates the current time and the current time less the UCMSleepTime parameter value (default value of 60 seconds).
   - [Last Update - SDQ] indicates 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].
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 systems that meet the following conditions:
   - They have the object defined in the Publish/Subscribe view.
   - They have a publish frequency set to Batch.
   - They have not expired.

Event Mode

Any system configured for real-time mode 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 Oracle Customer Hub (UCM) operations, such as Merge or Unmerge, or Link and Update, or Create New, and so on. These events are preconfigured by default, but are not published. However, Oracle Customer Hub (UCM) can be customized to prepare the outbound message (resulting from such operations) by using Oracle Customer Hub (UCM) Publish/Subscribe Service-EventPublishMethod.

> **Note:** You must customize the UCM application before you can publish Unmerge events. This is not part of standard UCM functionality and is not within the scope of Oracle Support Services. You must contact Oracle Consulting Services for help in implementing this customization.

About Outbound PROXY Services

Oracle Customer Hub (UCM) can publish message in XML or JSON format, based on *Message Type* specified for each registered system. For systems with SOAP *Message Type*, UCM publishes the message in XML format; for systems with JSON *Message Type*, UCM publishes the message in JSON format.

In middleware mode, if you are configuring UCM to consume an external web service (or REST service) from the middleware systems, the systems registered as middleware must use PROXY *Protocol Type,* the proxy business service and business service method supplied in System Registrations view; otherwise, they should have the transport information supplied.

For more information on how to create proxy business service from Siebel Outbound Web Services refer to section on *Configuring Publish and Subscribe Functionality*.

Configuring System Publish and Subscribe for Oracle Customer Hub (UCM)

The Publish and Subscribe infrastructure provides workflow policies, workflow processes, and Oracle Customer Hub (UCM) messages to publish data records that have been inserted or updated in Oracle Customer Hub (UCM). The workflow policies track changes to records within Oracle Customer Hub (UCM) 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. Use the following task to configure publish and subscribe.

To configure publish and subscribe for a system

1. Navigate to the Administration - Universal Customer Master screen, then the System Registrations view.
2. In the System Registrations list, select the system of interest, and then click the link in the System ID field.
3. From the System Detail view, select the Publish/Subscribe tab.
4. In the Publish/Subscribe view, select an existing record that you want to configure, or click New.
5. Enter the appropriate information in the available fields to define the publish and subscribe information for each object in the system.

   Use the following table for information on each of the fields.

Field	Description
Object Name	An object stored in Oracle Customer Hub (UCM) like Contact, Account, Household, and so on.
Publish Frequency	How often Oracle Customer Hub (UCM) update the system's information for the selected object. Select the 'Real Time' value for publish and subscribe functionality in real-time mode. Select the 'Daily Batch' value for publish in batch mode.

Field	Description
Last Published	The last time Oracle Customer Hub (UCM) published record changes for the selected object to this selected system.
Start Date	The start date of the publish and subscribe functionality.
End Date	The end date of the publish and subscribe functionality.
Event	Signifies if the object subscribes to event publishing.
Business Service	The business service being used for middleware system with PROXY protocol.
Business Service Method	The business service method for middleware system with PROXY protocol.
Comment	Comments on the publish and subscribe definition for the selected object of the selected system.

6. For middleware systems, configure to use PROXY *Protocol Type,* click on the *Business Service* field to select the proxy business service will pop-up an appropriate dialog box depending on the value in *Message Type* field:
    - For JSON Message Type, select REST Outbound Service
    - For SOAP Message Type, select Outbound Web Service
7. Step off the record to save the publish and subscribe functionality.

Related Books:

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

Modified Table for Oracle Customer Hub (UCM) Version 26.3

S_CIF_EXT_SYST Table

The following table shows details of the S_CIF_EXT_SYST table which is used for registered systems.

Column Name	Data Type	Comments
PAR_ROW_ID	Varchar (15)	Parent middleware Row Id

9 Configuring Data Decay Functionality in Oracle Customer Hub (UCM)

Configuring Data Decay Functionality in Oracle Customer Hub (UCM)

This chapter describes how to configure data decay in Oracle Customer Hub (UCM). It includes the following topics:

- *About Data Decay Functionality*
- *Configuring Data Decay Functionality*
- *Administering Data Decay Functionality*

About Data Decay Functionality

Data decay is the way in which managed information becomes degraded or obsolete. You manage data decay in contact and account records from the Data Decay dashboard. The following two factors cause data decay:

- **Data management is not up-to-date.** When a data management module is unable to catch up with data changes or suffers from a lack of proper attention.
- **Data storage application does not respond in time.** When the data storage application does not respond in time to a new standard.

How Is Data Decay Managed?

You use the following methods to manage data decay:

- **Decay metrics.** These are a set of configurable criteria that allows you to determine how stale data has become during a period. The decay metrics include:
  - **Decay indicator.** Indicates a numeric code, ranging from 0-100. Over a period, this number is counted down. The countdown is based on a set of configurable rules when updates do not occurs.
  - **Decay last updated date.** The initial value for the Decay Last Updated Date field is set by the record's Last Updated Date entry. Subsequently, the attribute and relationship level decay updated date is captured at real-time and set for the Decay Last Updated Date.
- **Decay granularity.** Decay granularity is addressed through *attribute decay*. With attribute decay, decay metrics are calculated based on an individual attribute.
- **Decay management.** Decay management can be divided into the following main components:
  - **Decay detection.** Captures updates to the monitored attributes and relationships of a record and configures the metrics
  - **Decay metrics recalculation.** Indicates a process to retrieve the decay metrics for the monitored attributes and relationships of a record. It uses a set of predefined rules to calculate the new metrics value and update the metrics.

- **Decay correctness.** Identifies the stale data. The identification is based on certain criteria and this component triggers a defined action.
- **Decay report.** Generates the Decay Metrics reports regularly.

Configuring Data Decay Functionality

The following topics describe how to configure data decay functionality for Oracle Customer Hub (UCM):

- *Activating Data Decay Run-Time Events*
- *Configuring the NetChart Server*
- *Additions and Enhancements for the Data Decay Functionality*

Activating Data Decay Run-Time Events

Data decay run-time events are inactive by default, so you must first activate them. Use the following task to activate data decay run-time events.

To activate data decay run-time events

1. In the Siebel Web Client, choose Administration-Runtime Events, Action Sets.
2. In the Action Sets view, query for the data decay action sets by entering the following search specification: *Decay*.
3. For each data decay action set, click the Active flag, then from the application-level menu, choose Reload Runtime Events.

Configuring the NetChart Server

To enable data decay functionality, you must have installed the Siebel edition of NetChart server and set the server parameters. Following the successful installation and setup, perform the following task to make sure that certain parameters are set.

To configure the NetChart server

1. In the Siebel Server installation, change the Siebel Server parameter.
2. Update the following ServerDataSrc named subsystem parameters:

    - **DSChartServer.** Set this value to the *ChartServerInstallMachineName* with the port number set to the default of 8001.
    - **DSChartImageFormat.** Set this value to PNG.

Additions and Enhancements for the Data Decay Functionality

This topic includes information about the additions and enhancements to the data decay object. It includes the following information:

- *New or Enhanced Screens for the Data Decay Functionality*
- *New or Enhanced Views and Applets for Data Decay Functionality*

New or Enhanced Screens for the Data Decay Functionality

The following information lists the new or modified screens for data decay functionality. The UI controls are New, Delete, and Query buttons.

New or Modified	Screen	Comments
Modified	CIF Administration Screen	UCM Account Data Decay Dashboard was added. UCM Contact Data Decay Dashboard was added.

New or Enhanced Views and Applets for Data Decay Functionality

The following information lists the new and enhanced views and applets for data decay functionality.

New or Modified	View	Applet
New	UCM Contact Data Decay Dashboard	UCM Decay Contact List Applet
		UCM Contact Work Phone # Decay Gantt Chart
		UCM Contact Home Phone # Decay Gantt Chart
		UCM Contact Cellular Phone # Decay Gantt Chart
		UCM Contact Email Address Decay Gantt Chart
New	UCM Account Data Decay Dashboard	UCM Decay Account List Applet
		Account Main Phone Number Decay Gantt Chart
		UCM Account Main Email Address Decay Chart Applet
		UCM Account Main Email Address Decay Gantt Chart

New or Modified	View	Applet
New	UCM Contact Decay Management View	UCM Contact Home Phone # Decay Chart Applet
		UCM Contact Work Phone # Decay Chart Applet
		UCM Contact Cellular Phone # Decay Chart Applet
		UCM Contact Email Address Decay Chart Applet
New	UCM Account Decay Management View	UCM Account Main Phone Numbers Decay Chart Applet
		UCM Account Main Fax Numbers Decay Chart Applet
		UCM Account Main Email Address Decay Chart Applet

Administering Data Decay Functionality

The following topics describes how to administer data decay functionality in Oracle Customer Hub (UCM):

- *Managing Data Decay Functionality*
- *Data Decay Action Sets for the Account Object*
- *Data Decay Action Sets for Contact*
- *Account Run-Time Events*
- *Contact Run-Time Events*

Managing Data Decay Functionality

This topic gives the following example of how Oracle Customer Hub (UCM) might be used. You might use Oracle Customer Hub (UCM) differently, depending on your business model.

Data decay functionality allows a data steward to access Contact and Account address records that have not changed over a set span of time. A data steward is responsible for deployed customer information in Oracle Customer Hub (UCM). After logging in the data steward receives an alert notification that a number of records have become stale. The records in question are three Account records and two Contact records. For accounts A, B, and C, the Account Main Phone Number and Account Email Address values have changed. Also two contact records have email addresses that have changed.

Data Decay Action Sets for the Account Object

The following topics contain information on data decay functionality action sets for the account object:

- *Action Set: UCM Account Decay New Record*
- *Action Set: UCM Account Decay Pre Write Record*
- *Action Set: UCM Account Decay Set Main Email Address*
- *Action Set: UCM Account Decay Set Main Fax Number*
- *Action Set: UCM Account Decay Set Main Phone Number*
- *Action Set: UCM Account Decay Update Record*

Action Set: UCM Account Decay New Record

The following table contains information on the UCM Account Decay New Record action.

Action Name	Action Type	Sequence	Business Service Name	Business Service Method	Business Service Context
UCM Account Decay New Set Field	BusService	1	UCM Data Decay Service	SetDecayMetrics	Primary Address ID: Primary CUT Address

Action Set: UCM Account Decay Pre Write Record

The following table contains information on the UCM Account Decay Pre Write Record action.

Action Name	Action Type	Sequence	Business Service Name	Business Service Method	Business Service Context
UCM Account Decay Pre Write Record	BusService	1	UCM Data Decay Service	CreateDecayFieldMap	Primary Address ID

Action Set: UCM Account Decay Set Main Email Address

The following table contains information on the UCM Account Decay Set Main Email Address action.

Action Name	Action Type	Sequence	Business Service Name	Business Service Method	Business Service Context
UCM Account Decay Set Main Email Address	BusService	1	UCM Data Decay Service	CreateDecayFieldMap	None

Action Set: UCM Account Decay Set Main Fax Number

The following table contains information on the UCM Account Decay Set Main Fax Number action.

Action Name	Action Type	Sequence	Business Service Name	Business Service Method	Business Service Context
UCM Account Decay Set Main Fax Number	BusService	1	UCM Data Decay Service	CreateDecayFieldMap	None

Action Set: UCM Account Decay Set Main Phone Number

The following table contains information on the UCM Account Decay Set Main Phone Number action.

Action Name	Action Type	Sequence	Business Service Name	Business Service Method	Business Service Context
UCM Account Decay Set Main Phone Number	BusService	1	UCM Data Decay Service	CreateDecayFieldMap	None

Action Set: UCM Account Decay Update Record

The following table contains information on the UCM Account Decay Update Record action.

Action Name	Action Type	Sequence	Business Service Name	Business Service Method	Business Service Context
UCM Account Decay Update Set Field	BusService	1	UCM Data Decay Service	SetDecayMetrics	Primary Address Id: Primary CUT Address

Data Decay Action Sets for Contact

The following topics contain information on data decay action sets for the contact object:

- *Action Set: UCM Contact Decay New Record*
- *Action Set: UCM Contact Decay Pre Write Record*
- *Action Set: UCM Contact Decay Set Cell Phone #*
- *Action Set: UCM Contact Decay Set Email Address*
- *Action Set: UCM Contact Decay Set Home Phone #*

- *Action Set: UCM Contact Decay Set Work Phone #*
- *Action Set: UCM Contact Decay Update Record*

Action Set: UCM Contact Decay New Record

The following table contains information on the UCM Contact Decay New Record action.

Action Name	Action Type	Sequence	Business Service Name	Business Service Method	Business Service Context
UCM Contact Decay New Set Field	BusService	1	UCM Data Decay Service	SetDecayMetrics	Primary Personal Address Id: Primary Personal Address

Action Set: UCM Contact Decay Pre Write Record

The following table contains information on the UCM Contact Decay Pre Write Record action.

Action Name	Action Type	Sequence	Business Service Name	Business Service Method	Business Service Context
UCM Contact Decay Pre Write Record	BusService	1	UCM Data Decay Service	CreateDecayFieldMap	Primary Personal Address Id

Action Set: UCM Contact Decay Set Cell Phone

The following table contains information on the UCM Contact Decay Set Cell Phone # action.

Action Name	Action Type	Sequence	Business Service Name	Business Service Method	Business Service Context
UCM Contact Decay Set Cell Phone #	BusService	1	UCM Data Decay Service	CreateDecayFieldMap	None

Action Set: UCM Contact Decay Set Email Address

The following table contains information on the UCM Set Email Address action.

Action Name	Action Type	Sequence	Business Service Name	Business Service Method	Business Service Context
UCM Contact Decay Set Email Address	BusService	1	UCM Data Decay Service	CreateDecayFieldMap	None

Action Set: UCM Contact Decay Set Home Phone

The following table contains information on the UCM Contact Decay Set Home Phone # action.

Action Name	Action Type	Sequence	Business Service Name	Business Service Method	Business Service Context
UCM Contact Decay Set Home Phone #	BusService	1	UCM Data Decay Service	CreateDecayFieldMap	None

Action Set: UCM Contact Decay Set Work Phone

The following table contains information on the UCM Contact Decay Set Work Phone # action.

Action Name	Action Type	Sequence	Business Service Name	Business Service Method	Business Service Context
UCM Contact Decay Set Work Phone #	BusService	1	UCM Data Decay Service	CreateDecayFieldMap	None

Action Set: UCM Contact Decay Update Record

The following table contains information on the UCM Contact Decay Update Record action.

Action Name	Action Type	Sequence	Business Service Name	Business Service Method	Business Service Context
UCM Contact Decay Update Set Field	BusService	1	UCM Data Decay Service	SetDecayMetrics	Primary Personal Address Id: Primary Personal Address

Account Run-Time Events

The following table contains information on data decay run-time events for the account object.

Sequence	Object Name	Event	Subevent	Conditional Expression	Action Set Name
1	Account	SetField Value	Main Phone Number	None	UCM Account Decay Set Main Phone Number
1			Main Fax Number	None	UCM Account Decay Set Main Fax Number
1			Main Email Address	None	UCM Account Decay Set Main Email Address
2	Account	WriteRecordNew	None	GetProfileAttr('UCMDecayField Updated:Account')="Y"	UCM Account Decay New Record
3	Account	PreWriteRecord	None	None	UCM Account Decay Pre Write Record
3	Account	WriteRecordUpda	None	GetProfileAttr('UCMDecayField Updated:Account')="Y"	UCM Account Decay Update Record

Contact Run-Time Events

The following table contains information on data decay run-time events for the account object.

Sequence	Object Name	Event	Subevent	Conditional Expression	Action Set Name
1	Contact	SetField Value	Work Phone #	None	UCM Contact Decay Set Work Phone #
1	Contact	SetField Value	Cellular Phone #	None	UCM Contact Decay Set Cell Phone #

Sequence	Object Name	Event	Subevent	Conditional Expression	Action Set Name
1	Contact	SetField Value	Home Phone #	None	UCM Contact Decay Set Home Phone #
1	Contact	SetField Value	Email Address	None	UCM Contact Decay Set Email Address
4	Contact	WriteRecordNew	None	GetProfileAttr('UCMDecayField Updated:Contact')="Y"	UCM Contact Decay New Record
3	Account	PreWriteRecord	None	None	UCM Contact Decay Pre Write Record
2	Account	WriteRecordUpda	None	GetProfileAttr('UCMDecayField Updated:Contact')="Y"	UCM Contact Decay Update Record

10 Configuring Oracle Customer Hub (UCM) Privacy Management

Configuring Oracle Customer Hub (UCM) Privacy Management

This chapter describes the configuration requirements and basic administrative tasks for Oracle Customer Hub (UCM) Privacy Management. It includes the following topics:

- *About Oracle Customer Hub (UCM) Privacy Management*
- *Process of Configuring Privacy Management*
- *Administering Privacy Management*
- *About HaleyAuthority in Oracle Customer Hub (UCM) Privacy Management*
- *About the Business Rules Architecture in Oracle Customer Hub (UCM) Privacy Management*
- *Privacy Vocabulary*
- *Modules with Privacy Rules*
- *Customization Methods*
- *About Blank Privacy Source Data and History Records*

About Oracle Customer Hub (UCM) Privacy Management

Oracle Customer Hub (UCM) Privacy Management provides an infrastructure that allows companies to implement privacy management solutions for their customer data. These solutions help companies to protect personal customer data and to comply with government privacy regulations.

Privacy vocabulary and sample rules included with Oracle Customer Hub (UCM) Privacy Management can be tailored to meet your organization's requirements. Oracle Customer Hub (UCM) Privacy Management comes preconfigured with a vocabulary and sample rules designed to meet the requirements of financial services companies doing business in the United States. It also includes concepts that are useful to businesses managing compliance with fair information practices including notice and choice as well as global accepted privacy principles such as use limitation (purpose), disclosure (recipients), data collection limitation (data categories), and others.

In particular, Oracle Customer Hub (UCM) Privacy Management business rules environment enables business analysts to model, test, and deploy rules that manage privacy information stored in Oracle Customer Hub (UCM) related to financial accounts and contacts.

Note: Use this chapter with *Siebel Business Rules Administration Guide* version 8.1.

Oracle Customer Hub (UCM) Privacy Management implementation requires the following components:

- **Oracle Customer Hub (UCM).** It maintains the privacy preferences.
- **An embedded rules engine.** It allows rules creation and processing. For background information on the rules engine, see *Siebel Business Rules Administration Guide* , version 8.1.

About Privacy Management Features and Capabilities

Oracle Customer Hub (UCM) Privacy Management solution consists of the following features and capabilities:

- **Privacy Data Model.** Provides preconfigured, privacy-data definitions to capture the following customer data :sharing consent, solicitation choices, election to receive privacy notices, and other necessary data to generate compliance reports and revision history.
- **Privacy Views.** Provide privacy data status and history views at the account, contact, financial account, and financial account contact level. The history views allow for compliance monitoring of updates to the customer's privacy preferences.
- **Financial Account Views.** Support a single view of financial accounts. Financial accounts include a data synchronization business service, UCM FINCORP Account Source Data and History Service, which stores data (as integration object instances) into the database.
- **UCM Privacy Workflows.** Administer privacy changes associated with contacts on a financial account. For more information on the privacy management workflows, see *About Privacy Management Workflow Processes*.
- **UCM Privacy Management Vocabulary.** Provides preconfigured, privacy-rule building blocks, also known as privacy vocabularies, and a material set of sample privacy rules. The vocabulary has modeled privacy entities and attributes and is built in the context of various privacy regulations. The sample rules can be used to create custom rules that are specific to a company's privacy policies.
- **Financial Account Integration Web Services.** Provides Web services to interact with systems across the enterprise. These Web services are enhanced to include privacy integration objects to query, update, and insert customers' privacy data in Oracle Customer Hub (UCM).

About Privacy Management Configuration

Oracle Customer Hub (UCM) Privacy Management solution requires you verify any configurations you make to ensure the privacy features work in Oracle Customer Hub (UCM). Configurations are necessary for Oracle Customer Hub (UCM), Siebel Business Rules Developer, and integration technologies. For information on configuring Oracle Customer Hub (UCM) Privacy Management, see *Process of Configuring Privacy Management*.

About Privacy Management Administration

Oracle Customer Hub (UCM) Privacy Management is administered from the Oracle Customer Hub (UCM) user interface as well as Siebel Business Rules Developer, which is based on HaleyAuthority from Haley Systems, Inc. Privacy rule authoring and administration occur only in Siebel Business Rules Developer. For information on creating privacy rules, see *Siebel Business Rules Administration Guide* , version 8.1.

Overview of Requirements for Privacy Management

Before configuring Oracle Customer Hub (UCM) Privacy Management, make sure the following preliminary software requirements are met:

- Oracle Customer Hub (UCM) must be installed.
- Siebel Business Rules Developer must be installed.

  > **Note:** Siebel Business Rules Developer is part of Siebel Tools installation. For more information, see *Siebel Installation Guide* .

- Siebel Business Rules Developer is required for design time of privacy rules but is not required in run time.

About Oracle Customer Hub (UCM) Rules Engine Helper

- Oracle Customer Hub (UCM) Rules Engine Helper is a prebuilt helper business service for master data to help preparing input for the rules engine and processing output from the rules engine. To review specific details on business services, including user properties, methods, and method arguments, use Siebel Tools. For information on using Siebel Tools, see *Using Siebel Tools* on the *Siebel Bookshelf* .

About Privacy Management Workflow Processes

Oracle Customer Hub (UCM) Privacy Management consists of the following preconfigured Privacy workflow processes. For background information on workflow processes, see *Siebel Business Process Framework: Workflow Guide* .

The following workflows and the corresponding sample privacy rules are provided to administer privacy changes associated with contacts on a financial account. Similar workflows and privacy rules can be created in Oracle Customer Hub (UCM) and Siebel Business Rules Developer. The workflows and privacy rules correspond to the privacy changes associated with prospects, consumers, companies, organizations, or households. Run-time events are used to trigger these workflows. The UCM Privacy Workflow processes are:

- *About UCM Privacy Process Workflow*
- *About UCM Privacy Process-Pending Expired*
- *About UCM Privacy Process-Reevaluate Privacy Batch*

About UCM Privacy Process Workflow

This workflow process is invoked by run-time events through the business component user property Named Method and can be configured to support additional events. This workflow handles the following events:

- **Financial account type changes.** The workflow is triggered, by way of run-time events, when a financial account type changes. For example, a car loan has changed from a consumer car loan to a business car loan because the car is now used as a company vehicle. Some privacy regulations mandate different compliance requirements for consumers than for businesses. The sample rules provide examples for both business

financial accounts, whose opt-in data sharing is less restrictive, and consumer financial accounts where data sharing is more restrictive.

- **New financial account contact has been added.** The workflow is triggered, by way of run-time events, typically during a financial account new booking process. In this process, a new financial account with contact information is submitted to Oracle Customer Hub (UCM) from an external system, such as a financial loan-origination system. The workflow calls the rules engine to set the initial privacy preference values for each contact and for the financial account to which the contact belongs. The workflow is also triggered if a new financial contact is added. The sample privacy rules written in Siebel Business Rules Developer determine the contact's privacy preferences based on the entry in contact's Primary Garage Address State field. The default privacy preference values stored in the privacy rule must be changed, with the changes based on the privacy regulations and the company's data sharing practice.
- **Privacy setting for a financial account contact has been updated.** The workflow is triggered, by way of run-time events, when a financial account contact's privacy setting is changed.
- **Financial account contact status has been set to Inactive.** The workflow is triggered, by way of run-time events, when a contact changes its status from Active to Inactive. When a contact's status changes, its privacy sharing rights might change, too. The sample privacy rules provide the logic to set the updated, privacy-preference values for the inactive contact, as well as reevaluate the privacy status of the financial account with which the inactive contact is associated.
- **Financial account contact garage address has been changed.** The workflow is triggered, by way of run-time events, when a contact on a financial account changes his or her address. The current U.S. privacy regulation allows states to pass their own privacy regulation and therefore different privacy rules and business logic might be applied depending on the state where the financial account belongs. For example, California Financial Information Privacy Act (also known as CA SB1) requires that financial institutions provide California-based residents with a special disclosure and allow a 45-day initial waiting period before sharing of data occurs. This initial waiting period might be different for other states. The sample Haley Privacy Knowledge Base provides examples of such logic.

About UCM Privacy Process-Pending Expired

Different states in the U.S. can mandate different initial waiting periods during which a Contact might elect to opt-in or opt-out of sharing his or her personal information. The workflow process can be invoked using a workflow process batch manager server component. It can be run in the background with a search specification to collect all the financial accounts where the pending period has expired. The duration of the pending period is specified as rules and is set in Oracle Customer Hub (UCM) when the financial account is created or updated. The pending period is preconfigured to 50 days. For information on how to run the workflow process batch manager server component, see *Siebel System Administration Guide* .

About UCM Privacy Process-Reevaluate Privacy Batch

This workflow process can be invoked using a workflow process manager server component. It is a self-looping workflow process which allows you to avoid query overhead in the workflow batch process manager. This workflow is triggered when there is a change to the privacy global settings, for example, some states' law changes the default privacy settings for sharing telemarketing data. The sample rules provide some basic logic to reevaluate telemarketing data sharing for the financial account and the contact. For background information on how to run workflow process manager server component, see *Siebel System Administration Guide* .

Process of Configuring Privacy Management

Perform the following tasks to configure Oracle Customer Hub (UCM) Privacy Management functionality. For more information on Oracle Customer Hub (UCM) Privacy Management, see *About Oracle Customer Hub (UCM) Privacy Management*.

1. Make sure UCM Privacy Workflow processes are available or deployed. For information on this task, see *Deploying Privacy Management Workflow Processes*.
2. Make sure all applicable user properties are set. For more information, see *Setting Privacy-Related User Properties*.

Deploying Privacy Management Workflow Processes

To deploy a workflow process, see the task in *Siebel Business Process Framework: Workflow Guide* . For background information on the UCM Privacy workflow processes, see *About Privacy Management Workflow Processes*.

This task is a step in *Process of Configuring Privacy Management*.

Setting Privacy-Related User Properties

This task is a step in *Process of Configuring Privacy Management*. Use this task to set privacy user properties.

To set privacy-related user properties

- Set the user properties in the following business components, using Siebel Tools:
  - *UCM FINCORP Account Contact Privacy*
  - *FINCORP Account*
  - *UCM FINCORP Account Contact Address*
  - *UCM FINCORP Account Contact Address Source Data and History*
  - *FINCORP Account Contact*

UCM FINCORP Account Contact Privacy

The following information lists the user properties that must be set for the UCM FINCORP Account Contact Privacy business component.

Name	Value	Comments
Named Method 1	"AccountContactPrivacyChanged", "INVOKESVC", "UCM FINCORP Account Contact Privacy", "Workflow Process Manager", "RunProcess", "'ProcessName'", "'UCM Privacy Process'", "'RowId'", "[Financial Account Id]", "'Module Names'", "'Input interpretation, Initialization,Reevaluate privacy,Core Privacy Rules,Generate Output'"	Invokes the privacy workflow process to reevaluate the privacy changes.

Name	Value	Comments
Named Method 2	"GetAddress", "INVOKE", "UCM FINCORP Account Contact Privacy", "RefreshRecord"	Updates the current record. It is mainly for the UI.
Named Method 3	"AccountContactPrivacyNew", "INVOKESVC", "UCM FINCORP Account Contact Privacy", "Workflow Process Manager", "RunProcess", "'ProcessName'", "'UCM Privacy Process'", "'RowId'", "[Financial Account Id]", "'Module Names'", "'Input interpretation, Initialization,Reevaluate privacy,New Financial Account_New Contact Event,Core Privacy Rules,Generate Output'"	Invokes the privacy workflow process when a new contact privacy has been created.
PEND Date Rollup Parent Buscomp Name	UCM FINCORP Account Privacy	The parent business component name for rolling up the pending end date.
PEND Date Rollup Parent Field Name	Financial Account Id	The parent business component field, which serves as the parent ID field in PEND end date field when rolling it up.
PEND End Required Fields 1	Nonaffiliate Flag	A mandatory field when rolling up the PEND end date. If this field is still set to PEND, the Pending status does not end. The Pending status ends only when all the mandatory fields are not set to PEND status.
PEND End Required Fields 2	Affiliate Flag	
PEND End Required Fields 3	Channel Flag	
PEND End Required Fields 4	Telemarketing Flag	
PEND End Required Fields 5	Receive Annual Notice Flag	

FINCORP Account

The following information lists the user properties that must be set for the FINCORP Account business component.

Name	Value	Comments
Named Method 1	"AccountTypeChanged", "INVOKESVC", "FINCORP Account", "Workflow Process Manager", "RunProcess", "'ProcessName'", "'UCM Privacy Process'", "'RowId'", "[Id]", "'Module Names'", "'Input interpretation,Initialization,Account Type Change Event,Core Privacy Rules,Generate Output'"	Invokes the privacy workflow process when the account type has been changed.

UCM FINCORP Account Contact Address

The following information lists the user properties that must be set for the UCM FINCORP Account Contact Address business component.

Name	Value	Comments
Named Method 1	"GetPrivacy", "INVOKE", "UCM FINCORP Account Contact Address", "RefreshRecord"	Updates the current record.

UCM FINCORP Account Contact Address Source Data and History

The following information lists the user properties that must be set for the UCM FINCORP Account Contact Address Source Data and History business component.

Name	Value	Comments
Named Method 1	"GetFinancialAccount", "INVOKE", "UCM FINCORP Account Contact Address Source Data and History", "RefreshRecord"	Updates the current record.
Named Method 2	"AccountContactAddressChanged", "INVOKESVC", "UCM FINCORP Account Contact Address Source Data and History", "Workflow Process Manager", "RunProcess", "'ProcessName'", "'UCM Privacy Process'", "'RowId'", "[Financial Account Id]", "'Module Names'", "'Input interpretation,Initialization,Consumer Address Change Event,Core Privacy Rules,Generate Output'"	Invokes the privacy workflow process when the garage address has been changed.

FINCORP Account Contact

The following information lists the user properties that must be set for the FINCORP Account Contact business component.

Name	Value	Comments
Named Method 2	"ContactInactivated", "INVOKESVC", "FINCORP Account Contact", "Workflow Process Manager", "RunProcess", "'ProcessName'", "'UCM Privacy Process'", "'RowId'", "[Asset Id]", "'Module Names'", "'Contact Inactivated Event'"	Invokes the privacy workflow process when a financial account has been deactivated.

Administering Privacy Management

Data stewards or administrators can monitor the privacy options for Contacts, Accounts, and Financial Accounts in the Oracle Customer Hub (UCM) user interface. The primary administrative tasks of authoring privacy rules for Privacy Management are undertaken in Siebel Business Rules Developer. For more information, see *Siebel Business Rules Administration Guide* .

Managing Privacy Data

Oracle Customer Hub (UCM) Privacy Management provides preconfigured privacy tabs in Oracle Customer Hub (UCM) screens. From these tabs, administrators or data stewards can monitor the privacy elections and history of individual contacts, accounts, financial accounts, or financial account contacts. For example, they can see if a customer has opted out of a marketing campaign.

Managing Privacy Rule Modules

Oracle Customer Hub (UCM) provides seed privacy rule modules. Use the following task to display privacy rule modules.

To display privacy rule modules

- Navigate to Administration - Business Rules.

  The Business Rules Knowledge Base is displayed. A window enables you to choose the knowledge base, and the Rule Module list enables you to choose a rule module from the chosen knowledge base.

Managing Privacy Run-Time Events

Oracle Customer Hub (UCM) privacy workflow process can be triggered through run-time events. Run-time events in the business component level are used in privacy management to invoke a workflow process that communicates with the rules engine to get privacy settings. Run-time events do the condition checking for different business components (entities) and trigger the specified actions when the conditions are met.

To view privacy-related run-time events

1. Navigate to Administration - Runtime Events, and perform the following query:
   a. In the Object Field choose Business Component.
   b. In the Action Set Field, enter UCM*

Run-Time Events for the Default Settings in Privacy Management

The following tables provide details of the run-time events used in the default privacy management settings.

The following information lists the Retrieve Initial Privacy Setting run-time events.

Sequence	Object Name	Event	Conditional Expression	Action Set Name
1	UCM FINCORP Account Contact Privacy	WriteRecordNe	`(GetProfileAttr('Haley Message') IS NULL OR GetProfileAttr('Haley Message')<>'Y')`	UCM Set FACPrivacyNew To Y And Refresh Record

Sequence	Object Name	Event	Conditional Expression	Action Set Name
2	UCM FINCORP Account Contact Privacy	WriteRecordNe	GetProfileAttr('FACPrivacyNew')='Y' AND [Address Id] IS NOT NULL	UCM New Financial Account and Financial Account Contact

The following table lists the initial privacy settings where the action set is “UCM Set FACPrivacyNew To Y And Refresh Record" with Actions.

Name	Action Type	Sequence	Description
UCM Set Profile Attribute	Attribute Set	1	Values are: • The profile attribute value is FACPrivacyNew. • The set operator value is Set. • The value field value is Y.
UCM Refresh Record	Invoke Method	2	The method name is GetAddress.

The following table lists the initial privacy settings where the action set is “UCM New Financial Account and Financial Account Contact" with actions.

Name	Action Type	Sequence	Description
UCM Set Profile Attribute	Attribute Set	1	Values are: • The profile attribute value is FACPrivacyNew. • The set operator value is Set. • The value field value is N.
UCM New Financial Account/ Financial Account Contact	Invoke Method	2	The method name is AccountContactPrivacyNew.
Refresh	Invoke Method	3	The method name is GetAddress.

The following table lists the Reevaluate Privacy Setting, run-time event.

Sequence	Object Name	Event	Conditional Expression	Action Set Name
1	UCM FINCORP Account Contact Privacy	WriteRecordUpdat	`GetProfileAttr('Haley Message') IS NULL OR GetProfileAttr('Haley Message')<>'Y`	UCM Update Financial Account Contact Privacy

The following table lists reevaluate privacy setting, where the action set is “UCM Update Financial Account Contact Privacy" with actions.

Name	Action Type	Sequence	Description
UCM Update Financial Account Contact Privacy	Invoke Method	1	The method name is AccountContactPrivacyChanged
Refresh	Invoke Method	2	The method name is GetAddress

The following table lists the Update Account Type, Run-Time Event.

Sequence	Object Name	Event	Subevent	Conditional Expression	Action Set Name
1	FINCORP Account	SetFieldValue	Type	`(GetProfileAttr('Haley Message') IS NULL OR GetProfileAttr('Haley Message')<>'Y')`	UCM Set FinancialAccountTypeChanged To Y
2	FINCORP Account	WriteRecordUpdate	Not applicable	GetProfileAttr('FinancialAccountTyp Changed')='Y	UCM Update Financial Account Type
1	FINCORP Account	WriteRecordNew	Not applicable	GetProfileAttr('FinancialAccountTyp Changed')='Y	UCM Set FinancialAccountTypeChanged To N

The run-time event on WriteRecordNew is used to suppress the Update Account Type action for a new financial account case.

The following table lists Update Account Type, where the action set is “UCM Set FinancialAccountTypeChanged To Y" with actions.

Name	Action Type	Sequence	Description
UCM Set Profile Attribute	Attribute Set	1	Values are: • The profile attribute value is FinancialAccountTypChanged. • The set operator value is Set. • The value field value Y.

The following table lists Update Account Type, where the action set is “UCM Update Financial Account Type" with actions.

Name	Action Type	Sequence	Description
UCM Set Profile Attribute	Attribute Set	1	Values are: • The profile attribute value is FinancialAccountTypChanged. • The set operator value is Set. • The value field value is N.
UCM Update Financial Account Type	Invoke Method	2	The method name is AccountTypeChanged.

The following table lists Update Account Type, where the action set is “UCM Set FinancialAccountTypeChanged To N" with actions.

Name	Action Type	Sequence	Description
UCM Set Profile Attribute	Attribute Set	1	Profile Attribute= FinancialAccountTypChanged; Set Operator=Set; Value="N"

The following table lists Address Change Update Privacy Setting, run-time event.

Sequence	Object Name	Event	Subevent	Conditional Expression	Action Set Name
1	UCM FINCORP Account Contact Address	SetFieldValue	Personal Address Id	`[Address Type] = LookupValue ('FINS_CF_ADDR_USE_TYPE', 'Garage') AND ([Contact Privacy Mod Number]<1 OR [Contact Privacy Mod Number] IS NULL)`	UCM Set GarageAddressChanging To Y
1	UCM FINCORP Account Contact Address	WriteRecordUpdate	Not applicable	`(GetProfileAttr('Haley Message') IS NULL OR GetProfileAttr('Haley Message')<>'Y') AND GetProfileAttr('GarageAddressCha`	UCM Refresh Record On FINCORP Account Contact Address
2	UCM FINCORP Account Contact Address	WriteRecordUpdate	Not applicable	`[Address Type] = LookupValue ('FINS_CF_ADDR_USE_ TYPE', 'Garage') AND (GetProfileAttr('Haley Message') IS NULL OR GetProfileAttr('Haley Message')<>'Y') AND [Contact Privacy Mod Number]>0`	UCM Set GarageAddressChanged To Y
1	UCM FINCORP Account Contact Address	WriteRecordNew	Not applicable	GetProfileAttr('GarageAddressChanged')='Y AND [UCM Type Code] = LookupValue('UCM_SDH_TYPE_MLOV', 'History')	UCM Update Financial Account Contact Address

Sequence	Object Name	Event	Subevent	Conditional Expression	Action Set Name
	Source Data and History				

The following table lists Address Change Update Privacy Setting action set “UCM Set GarageAddressChanging To Y" with actions.

Name	Action Type	Sequence	Description
UCM Set Profile Attribute	Attribute Set	1	• The profile attribute value is GarageAddressChanging • The set operator value is Set • The value field value is Y

The following table lists Address Change Update Privacy Setting action set “UCM Refresh Record On FINCORP Account Contact Address" with actions.

Name	Action Type	Sequence	Description
UCM Refresh Record	Invoke Method	1	The method name is GetPrivacy

The following table lists Address Change Update Privacy Setting action set “UCM Set GarageAddressChanged To Y" with actions.

Name	Action Type	Sequence	Description
UCM Set Profile Attribute	Attribute Set	1	• The profile attribute is GarageAddressChanged • The set operator is Set • The value field value is Y.

The following table lists Address Change Update Privacy Setting action set “UCM Update Financial Account Contact Address" with actions.

Name	Action Type	Sequence	Description
UCM Set Profile Attribute	Attribute Set	1	• The profile attribute is GarageAddressChanged • The set operator is Set • The value field value is N.
UCM Refresh Record	Invoke Method	2	The method name is GetFinancialAccount.
UCM Update Financial Account Contact Address	Invoke Method	3	The method name is AccountContactAddressChanged.

Name	Action Type	Sequence	Description

Sequence number 2 in the previous table is used to get the parent Financial Account Id and pass it to the workflow, so that it can query with this ID and construct the request message for the Rules Engine.

The following table lists run-time event for the Retrieve Privacy Setting when a contact is deactivated.

Sequence	Object Name	Event	Subevent	Conditional Expression	Action Set Name
1	FINCORP Account Contact	SetFieldValue	Status	`[Status] = LookupValue('CONTACT_STATUS', 'Inactive') AND (GetProfileAttr('Haley Message') IS NULL OR GetProfileAttr('Haley Message')<>'Y')`	UCM Set ContactInactivated To Y
1	FINCORP Account Contact	WriteRecordUpdate	Not applicable	GetProfileAttr('ContactInactivated')='Y	UCM Contact Inactivated

The following table lists Retrieve Privacy Setting when contact is deactivated action set with “UCM Set ContactInactivated To Y" with actions.

Name	Action Type	Sequence	Description
UCM Set Profile Attribute	Attribute Set	1	• The profile attribute value is ContactInactivated • The set operator value is Set • The value field value is Y

The following table lists Retrieve Privacy Setting when contact is deactivated action wet “UCM Contact Inactivated" with actions.

Name	Action Type	Sequence	Description
UCM Set Profile Attribute	Attribute Set	1	• The profile attribute value is ContactInactivated • The set operator value is Set • The value field value is N
UCM Contact Inactivated	Invoke Method	2	ContactInactivated

About HaleyAuthority in Oracle Customer Hub (UCM) Privacy Management

HaleyAuthority 5.2 enables users to develop new or customize existing business rules as well as the testing capabilities. It is integrated into Oracle Customer Hub (UCM) Privacy Management solution with two Siebel CRM plug-ins:

- **Object importer.** The object importer enables you to select a subset of the Siebel object model and import it into HaleyAuthority to create concepts and relations.
- **Deployment wizard.** The deployment wizard guides you when deploying a rule module and storing it in the run-time tables.

At run time, requests are made to the business rule to evaluate a single financial account and its related entities (contacts, addresses, privacy data, and so on) and the business rule responds with the results of its evaluation.

About the Business Rules Architecture in Oracle Customer Hub (UCM) Privacy Management

Company business rules and policies are developed and tested within HaleyAuthority. The sample business rules are organized by privacy related events. You might implement your business-specific rules in separate modules as well as tailor the existing sample rules. The conceptual role model contains the vocabulary of the privacy rules. It includes the following: the concepts, entities and values, the dictionary, the relations between concepts, the phrasing of those relations, the definitions of tables, and the standard instances of the concepts and relations, such as state codes, cover letter, and privacy notice type codes, and so on.

Privacy Vocabulary

Oracle Customer Hub (UCM) Privacy Management solution business rules are developed using English language sentences. These sentences are made using concepts (nouns), relations between concepts, phrasing for those relations (verbs, adverbs, adjectives, prepositions and sentence fragments), and table type definitions that have been predefined in the sample knowledge base. These form the conceptual role model. This topic describes the key components of the conceptual role model that you use and customize. The sample knowledge base is named, Siebel_UCM_PrivacyMgmt.akb, and it is in the following directory:

```
Tools/Rule/KnowledgeBase
```

For more information on customization, see *Customization Methods*.

About the Conceptual Role Model

The sample conceptual role model included with Oracle Customer Hub (UCM) Privacy Management solution was developed using input from several sources:

- Language used in privacy policies across various industries
- Privacy vocabulary research summarized the W3C's Platform for Privacy Preferences (P3P) project
- Terminology used by some U.S. financial services industry customers of Siebel CRM

The conceptual role model is intended to cover a broad set of U.S. financial services requirements as well as be a starting point for privacy-policy management for other industries and geographical areas. You must modify the conceptual role model to support the mapping of additional elements from Oracle Customer Hub (UCM), to customize instances of communication message types.

About Entities

Entities are concepts that have attributes. In Oracle Customer Hub (UCM) Privacy Management solution, there are two types of entities:

- **Generated entities from Siebel Object Importer.** They come directly from definitions in Siebel business components. The main generated entities are: FINCORP Account, FINCORP Account Contact, UCM FINCORP Account Contact Address, UCM FINCORP Account Contact Garage Address Source Data and History, UCM FINCORP Account Contact Privacy, and UCM FINCORP Account Privacy. For more information about these entities, see *Siebel Business Rules Administration Guide* , version 8.1.
- **Abstract entities.** They are built on other entities or value concepts for convenience when using the Authority. The abstract entities in Oracle Customer Hub (UCM) Privacy Management solution include:
  - **Abstract entity-privacy flag.** This abstract entity is a named collection of privacy attributes for which an option is selected. The default instances of privacy preferences reflect the standard elections for U.S. financial services companies: affiliate flag, telemarketing flag, nonaffiliate flag, and channel flag for dealer sharing. The privacy preferences can also be mapped to Platform for Privacy Preferences Project (P3P) privacy statements to maintain preferences that are consistent with P3P privacy policies.
  - **Abstract entity-privacy flag source.** This abstract entity is a choice that indicates whether a privacy flag is allowed, denied, or pending, and whether the privacy flag was elected by default or by request based on customer direction. Typically, this entity has the following values: Opt in default, Opt in request, Opt out default, Opt out request, Pending default, and Pending request.
  - **Abstract entity-state code.** This abstract entity indicates which state out of all the states that come with all the financial account contacts on the financial account is the most restrictive state for that financial account. Some states such as California, Puerto Rico, and Vermont increase the privacy regulatory requirements set by the Graham-Leach-Bliley Act (GLBA) or FCRA. They are listed as separate instances. The other states are listed as one generic instance. For any state code other than the fifty states or DC or Puerto Rico, this code is set to *XX*. Note that *XX* represents invalid states; that is, state codes that do not apply to the 50 U.S. states, DC, or Puerto Rico. This practice is common for managing U.S. state legislation and policy. For example, a customer can have one of the following practices:
    - One state code corresponds to one state.
    - One state code corresponds to all states under the same federal law such as GLBA law.

About Value Concepts

Value concepts are created by Siebel Object Importer, using field definitions of the Siebel object model business components. To simplify the handling of strings in tables, string instances generally have the same instance name and string value, by convention. For more information about Siebel Object Importer, see *Siebel Business Rules Administration Guide* .

Relations and Phrasings

Siebel Object Importer can create entity or value concepts, as well as establish the relations between these two and generate simple phrasings such as *an entity has a value,* and so on. Oracle Customer Hub (UCM) Privacy Management solution also contains a library of privacy-related complicated relations and phrasings.

Oracle Customer Hub (UCM) Privacy Tables

The following tables are used in Oracle Customer Hub (UCM) Privacy Management solution. The following table shows details of the Privacy Option with Source Ranking table.

Module	Standard Phrasing to Retrieve Values	Description
Privacy option with source hierarchy	A privacy flag source has a rank	This table maintains the relative priority of privacy flag sources when rolling up privacy elections to the financial account level. This table uses the entity, privacy flag source, and a manually created integer value concept, rank. **Note:** An opt-out request must always be ranked number 1.

The following table shows details of the State Code Cover Letter Table and State Code Notice Type table.

Module	Standard Phrasing to Retrieve Values	Description
CUST.1 Owner type tables	A state code with an owner type determines whether a cover letter or privacy notice is sent.	These tables store the cover letter and privacy notice type codes based on the state code and owner type, or communication message type.

The following table shows details of the Owner Type State Code Change Cover Letter Table.

Module	Standard Phrasing to Retrieve Values	Description
CUST.2 Address changes the tables	A cover letter type is determined after a second state code changes from a first state code.	These tables store the cover letter type codes based on the original state code and the new state code. These tables are instances of the same table type. The correct table is selected by the applicability condition on the parent module based on the owner type.

The following table shows details of the State Restrictiveness Ranking Table.

Module	Standard Phrasing to Retrieve Values	Description
U.S. country rules	A state has a rank.	This table ranks states by their restrictiveness and is used to determine which state must be the privacy state code for a financial account when there are multiple states that come with all the financial account contacts on the financial account.

The following table shows details of the Default Privacy for Privacy Flag table.

Module	Standard Phrasing to Retrieve Values	Description
U.S. FS.1 Default privacy option tables.	A state determines a privacy flag.	These tables determine the default privacy flag options for each privacy flag based on the state code.

Modules with Privacy Rules

The top-level modules are grouped by privacy events and the type of rules in the module. In addition, the modules are separated by the order of evaluation because only top-level modules can have priorities. The following are the major modules:

- *Account Type Change Event Module*
- *Consumer Address Change Event Module*
- *Contact Inactivated Event Module*
- *Core Privacy Rules Module*
- *Generate Output Module*
- *Initialization Module*
- *Input Interpretation Module*
- *New Financial Account_New Contact Event Module*
- *Pending Period Expired Event Module*
- *Reevaluate Privacy Module*

Account Type Change Event Module

This module has a priority of six. It contains the rules that apply when a FINCORP account type is changed from Consumer to Business or conversely.

Consumer Address Change Event Module

This module has a priority of eight. It contains the rules that apply when a FINCORP account contact of a consumer type of FINCORP account moves to a new state. Certain Oracle submodules are applicable only when this causes a change to the privacy state code for the FINCORP account.

Contact Inactivated Event Module

This module has a priority of zero. It contains the rules that apply when a FINCORP account contact is deactivated from a FINCORP account. The sample rules specific to Contact Inactivated Event are included in the Oracle submodule, Customer Rules. The other Oracle submodules are a copy of the core modules including: Input interpretation, Initialization, Core Privacy Rules, Generate Output and Reevaluate privacy. This is because Contact Inactivated Event is for both active and inactive contacts. Whereas the other events are for active contacts only. These events cannot share the same core modules.

Core Privacy Rules Module

This module has a priority of five. It contains the core rules that apply generally to all the evaluation requests:

- **CUST. Customer rules.** This module contains customer-specific core rules. The sample rule is for the customers which are designated as consumers. This sample rule forces the channel partner sharing preference elections to be identical to the nonaffiliate sharing preference elections of all consumers outside of California who have a channel partner type of Brand X or Brand Y.
- **U.S. FS. Evaluate core privacy rules.** This module contains the core rules that generally apply to all U.S. Financial Services customers for any evaluation request.
- **U.S. FS.1. Determine privacy state code.** This module contains statements that set the privacy state code values for a business or consumer account and determine whether it has changed from a previous value.
- **U.S. FS.2. Determine precedence of privacy options and sources.** This module contains statements that use the privacy option with the source hierarchy to determine the roll up of privacy preferences to the financial account level.
- **U.S. FS.3. Handle privacy notices and letters.** This module contains the statements that determine whether a contact receives an annual notice and what cover letter and notice must be sent to the customer, except in the case where the contact has moved.

Generate Output Module

This module has a priority of zero. It contains the rules that perform an evaluation after the main body of business rules is used to set the values on the privacy record generated for Oracle Customer Hub (UCM). The sample rules set the value for a concept only when the concept value must be changed following the evaluation of the main business rules.

Initialization Module

This module has a priority of 10. It contains business rules, including the definitions and tables of the default values:

- **CUST. customer rules.** This module contains initial business rules and table instances that are customer-specific. It includes table instances of the cover letter types and privacy notice types based on the privacy state code and owner type (communication message type) and the cover letter types based on the state address change and owner type.
- **Expired pending period.** This module contains a business rule to determine that a pending period has expired.
- **Initial values.** This module contains the initial business rules and table instances that are not country-specific or industry-specific. It includes the table instance establishing the hierarchy of the privacy option with the following sources: Opt In Request, Opt In Default, Pend, Opt Out Default, or Opt Out Request.
- **U.S. country rules.** This module contains the initial business rules and table instances that apply to U.S. companies (for example, a rule establishing the most restrictive state for each contact). It includes a table instance ranking the restrictiveness of the U.S. states. It contains some examples. Refer to your company's consumer privacy best practices, and tailor these samples rules accordingly.
- **U.S. FS. Financial Services industry rules.** This module contains the initial business rules and table instances that apply to U.S. financial services companies. These rules are most commonly managed in financial industries. However, depending on your company's asset and privacy policy, you might reuse these rules or tailor them to your needs.

  > **Note:** The FINCORP business component corresponds to S_ASSET, which is the underlying Siebel schema. If you want to reuse these rules for your industry-specific business component, see *Customization Methods*.

- **U.S. FS.1. default privacy option tables.** This submodule includes the tables of the default privacy options for each privacy flag. These tables are organized according to state code.

Input Interpretation Module

This module has a priority of 11. It contains the rules that interpret input data and determine the key privacy related facts:

- Annual notice requirement
- Contact delete
- Determine state where the garage address is located
- Determine primary FINCORP account contact
- Privacy state code

New Financial Account_New Contact Event Module

This module has a priority of nine. It contains the rules that apply when a new FINCORP account or a FINCORP account contact is added.:

- **Determine new financial account or new financial account contact.** This module contains the rules that decide on the event object type: whether it is for a new FINCORP account or for a new FINCORP account contact.
- **New contact.** This module contains the rules that apply for a new FINCORP account contact case. Business and consumer types of FINCORP account have different privacy regulations, which are further specified in two separate submodules.
- **New financial account.** This module contains the rules that apply for a new FINCORP account case.

Pending Period Expired Event Module

This module has a priority of zero. It contains the rules that apply when the pending period for the consumer type FINCORP account expires, such as setting the default privacy options for each privacy preference based on the privacy state code.

Reevaluate Privacy Module

This module has a priority of zero. It contains the statements that interpret and recognize the input privacy options with sources on FINCORP account contact privacy level. By executing rules in this module, all customer requested flag values are preserved and not limited only to OUT Requested.

Customization Methods

This topic describes the following customization methods:

- *Guidelines for Customizing the Sample Knowledge Base*
- *Required Customizations*
- *Examples of Customizing the Sample Knowledge Base*

Guidelines for Customizing the Sample Knowledge Base

This topic provides guidelines for customizing the sample knowledge base. The sample rules for Oracle Customer Hub (UCM) Privacy Management solution are intended to be a template for a customer-specific implementation of business policies. You can modify, simplify, translate, replace, or use them as a design model as dictated by customer needs.

However, by following the guidelines on customizing the sample knowledge base, the process of integrating future releases of Oracle Customer Hub (UCM) Privacy Management solution can be simplified.

- **Insert rules into the existing CUST customer specific modules in the sample knowledge base whenever practical.** The sample CUST modules contain rules and tables that either require customization for each customer or are intended to isolate customizations from the prepackaged solution. To the extent that these customizations involve rules specific to multiple jurisdictions, the jurisdiction prefix must be included in the modules.
- **Customize instances of rules in customer installations, such as owner type, FINCORP account type or customer type, and so on.** If the instances are different, then the changes can be done in the knowledge base only. To the extent that further modifications to entities or relations in the conceptual role model are required, then these modifications must be done in the Siebel Repository and copied to the knowledge base, using Siebel Object Importer. Siebel Object Importer does not remove the old concepts and relations, because it is better to identify them using the version custom property or a comment.
- **Modify the standard sample rules when necessary.** You must identify your own rules in the new customer-specific modules by adding an appropriate prefix. The modified or deleted sample rule can easily be retained for history as a draft rule by inserting unrecognized text at the beginning or by changing the effective date of the rule to be expired. In addition, using the custom version property provides an additional means of identifying your customization.

  By following these guidelines, you can identify the customizations if a customer wants to integrate the features of future releases of the knowledge base.

Required Customizations

The following customizations are mandatory for any new installation of Oracle Customer Hub (UCM) Privacy Management solution:

- *CUST. Customer Rules*
- *Owner Types and Cover Letter Types*

CUST. Customer Rules

The sample knowledge base has sample rules in this module that force the channel flag to be identical to the nonaffiliate flag for consumers who are customers of Brand X or Brand Y and live outside of California. You must customize these conditions to meet your business policies for their channel flags or other criteria or else delete these statements.

Owner Types and Cover Letter Types

You must determine what owner types to use, if any. Oracle Customer Hub (UCM) has the Brand X and Brand Y as sample owner type values. If the customer has an owner, channel partner, or dealer and these affect the version of the cover letters or privacy notices, then you must make these changes:

- Modify the instances of the owner type based on the values used in Oracle Customer Hub (UCM).

- Edit the State Code Cover Letter Table and State Code Privacy Notice Table column headings to match the new owner types. Modify the table entries to reflect the customer's cover letter type codes and privacy notice type codes.
- Edit the Oracle submodule names and State Change Cover Letter Table instance names in CUST.2 Address changes tables to reflect the new owner types. Add additional modules and table instances as necessary if more than two communication message types are used. Edit the table entries to reflect the appropriate cover letter type codes for state changes for your company.

If you have only a single version of the cover letters or privacy notices (other than state-specific versions), then you must do the following:

- Add an instance to the concept owner type with name set to standard and value set to “standard," and delete the Brand X and Brand Y.
- Edit the State Code Cover Letter Table and State Code Privacy Notice Table column headings to have a single communication message type set to standard. Modify the table entries to reflect the customer's cover letter type codes and privacy notice type codes.
- Change the submodule name of *CUST.2.1. Brand X tables* to *CUST.2.1. Standard tables,* and change the name of the *Brand X State Change Cover Letter Table* instance to *Standard State Change Cover Letter Table* instance. Edit the table entries to reflect the appropriate cover letter type codes for state changes for your company.

Examples of Customizing the Sample Knowledge Base

This topic gives examples of customizing the sample knowledge base. You might use this feature differently, depending on your business model. Review the following examples to see if they apply to your implementation:

- *Capturing Privacy Elections for a New Business Channel*
- *Changing the Address for Privacy Preferences*
- *Implementing a New Company Policy to Be Compliant with OFAC*
- *Modifying New State Default Privacy Options upon Pending Expiration*
- *Modifying the State Restrictiveness Ranking Table*
- *Using New Business Components in Oracle Customer Hub (UCM) to Reuse Modules or Rules in the Privacy Knowledge Base*

Capturing Privacy Elections for a New Business Channel

You want to capture privacy elections for a new channel added to the business, for example, Insurance.

Adding a new privacy preference involves several changes to the business rules after the privacy preference is added in the Siebel table and Siebel Repository. Use the following task to capture privacy elections for a new business channel.

To capture privacy elections for a new business channel

1. Run the Siebel Object Importer to bring the change into the knowledge base conceptual role model.

   New concepts and relations are generated automatically.
2. Add the instances for the privacy preference to the privacy preference concept Insurance.
3. Add a new table with the default privacy preference values for insurance, copying the format of the existing Default Privacy Option Tables.

4. Add the business rules for this new privacy preference.

Changing the Address for Privacy Preferences

You want to evaluate the privacy preferences based on the addresses other than Garage Address, for example, Permanent Address. In such a case, you must change the rules in both the Siebel CRM configuration and HaleyAuthority.

Note: If you do not require consistency in concept and entity naming, the required changes are minimal.

To change the address for privacy preferences (solution one)

1. In Oracle Customer Hub (UCM), add the LOV Permanent to the Address Type field for FINCOPR Account Contact Address.
2. Change the business component search specification, UCM FINCORP Account Contact Garage Address Source Data and History, from:
   `[Address Type] = 'Garage' AND [UCM Type Code] = 'History' AND [Parent Type] = 'Financial Account Contact'` to: `[Address Type] = 'Permanent' AND [UCM Type Code] = 'History' AND [Parent Type] = 'Financial Account Contact'`
3. Modify the run-time event to look for changes in Permanent Address instead of Garage Address.
4. In HaleyAuthority, search for the word *Address Type*, replace *Garage* with *Permanent* in all the statements found, then deploy the changed modules.

Changing the Address for Privacy Preferences (Solution Two)

If you require consistency in the wording of concept rules and entity rules in Oracle Customer Hub (UCM) and HaleyAuthority, then perform the following procedure.

To change the address for privacy preferences (solution 2)

1. Make the changes specified in solution one.
   For more information, see *Changing the Address for Privacy Preferences*.
2. Create a new business component.
3. Modify the business object and integration object definitions.
4. Modify the rules administration configuration.
5. Run Siebel Object Importer to bring the new concept and relations into HaleyAuthority.
6. Build the new rules, using the existing rules for garage address.
7. Create a new module to hold the new rules.

Implementing a New Company Policy to Be Compliant with OFAC

You pass a new company policy to be compliant with the Office of Foreign Assets (OFAC) list, and privacy rules must be created around OFAC. The existing field exists in both the Siebel Repository and Haley Conceptual model, so do not run Siebel Object Importer again.

Add a new submodule under the corresponding event module, and then add the relevant business logic.

Modifying New State Default Privacy Options upon Pending Expiration

You want the default privacy options to be set for each customer. The default privacy options are based on the privacy state code for each preference.

Modify table entries in the Default Privacy Option Table in the U.S. FS.1 Default privacy option tables. The entries for additional state regulations can be added by inserting additional rows in this table.

Modifying the State Restrictiveness Ranking Table

You want to alter the ranking of states used to determine the privacy state code for the financial account.

Modify the priority of the states in the State Restrictiveness Ranking Table in U.S. country rules or add additional states with priorities. If Puerto Rico has the same rank as another entry, Puerto Rico has a rule that makes it take precedence in the module, U.S. country rules.

Using New Business Components in Oracle Customer Hub (UCM) to Reuse Modules or Rules in the Privacy Knowledge Base

FINCORP is a standard business component and corresponds to S_ASSET. You want to reuse the rules related to the FINCORP account, but you must use your industry-specific business component based on S_ASSET.

Make these modifications:

- Run Siebel Object Importer to bring the new business components into knowledge base conceptual role model.

  New concepts and relations are generated automatically.
- Change the rules that you want to reuse to use the concepts and relations that you just imported. Or, copy the original rules instead of changing them.
- Reparse the rules.

About Blank Privacy Source Data and History Records

Blank privacy history records might be created in the database because external applications send financial account insert requests to Oracle Customer Hub (UCM). To evaluate financial account and financial account contact's privacy preferences, the external application is required to send a privacy record with the financial account and the financial account contact insert request. The source is captured in Oracle Customer Hub (UCM) Transaction Manager. If there is no privacy preference preselected, the privacy preferences in the source record are empty. This record in the source data and history (SDH) table is considered to be a blank record.

If you do not select the privacy preferences before sending information to the rules engine, then the blank record is captured as history. Even though there is no privacy preference in this history record, the record does have some other values internally, such as a foreign key to its parent record type. There are two reasons why this occurs:

- **S_PRIVACY is a separate table.** Creating a financial account (FA) or financial account contact (FAC) does not automatically create a child privacy record.
- **There is no NewRecord action available in the core rules engine to call back Oracle Customer Hub (UCM) to create a new privacy record.** A privacy record must be created before sending it to the rules engine even though it is a new Financial Account or Financial Account Contact event.

For these reasons, a empty record exists in the base table and the SDH table. The base table empty record exists for a short period because the rules engine is triggered. However, empty records in the SDH table remain.

If you want to remove these blank records, there are several possible solutions. You must evaluate the situation according to your deployment needs. Removing blank privacy SDH records might affect performance adversely.

One solution is to write a script in the business service, UCM FINCORP Account Source Data and History Service. When the Insert method is called, check the values for some of the required fields in UCM FINCORP Account Privacy Source Data and History, UCM FINCORP Account Contact Privacy Source Data and History. If the values are empty, delete the record. Use the following task to add a script to a business service.

To add a script to a business service

1. In Siebel Tools, find the business service, UCM FINCORP Account Source Data and History Service.
2. Right–click, and choose Edit Server Script.
3. In Service_InvokeMethod, add the following script in eScript:

   ```
   Program Language = JS
   ```
4. Save and compile the changes.

Example Code to Remove Blank Records

The following script removes blank records.

Note: You might have different criteria for what constitutes a blank record. Modify the following script accordingly.

```
function Service_InvokeMethod (MethodName, Inputs, Outputs)

{

 if (MethodName == "Insert")

 {

 var oBusObj = TheApplication().GetBusObject("UCM FINCORP Account Source Data and
History");

 var oBusCompFAP = oBusObj.GetBusComp("UCM FINCORP Account Privacy Source Data and
History");

 var oBusCompFACP = oBusObj.GetBusComp("UCM FINCORP Account Contact Privacy Source
Data and History");
 with (oBusCompFAP)
 {
 ClearToQuery();
 ActivateField("Affiliate Flag");
```

```
ActivateField("Nonaffiliate Flag");
ActivateField("Telemarketing Flag");
ActivateField("Channel Flag");
SetSearchSpec("Affiliate Flag", "IS NULL");
SetSearchSpec("Nonaffiliate Flag", "IS NULL");
SetSearchSpec("Telemarketing Flag", "IS NULL");
SetSearchSpec("Channel Flag", "IS NULL");
ExecuteQuery(ForwardBackward);
while(FirstRecord())
{
DeleteRecord();
NextSelected();
}
}
with (oBusCompFACP)
{
ClearToQuery();
ActivateField("Affiliate Flag");
ActivateField("Nonaffiliate Flag");
ActivateField("Telemarketing Flag");
ActivateField("Channel Flag");
SetSearchSpec("Affiliate Flag", "IS NULL");
SetSearchSpec("Nonaffiliate Flag", "IS NULL");
SetSearchSpec("Telemarketing Flag", "IS NULL");
SetSearchSpec("Channel Flag", "IS NULL");
ExecuteQuery(ForwardBackward);
while(FirstRecord())
{
DeleteRecord();
NextSelected();
}
}
}
}
```

11 Configuring UCM Transaction Manager Callback Functions

Configuring UCM Transaction Manager Callback Functions

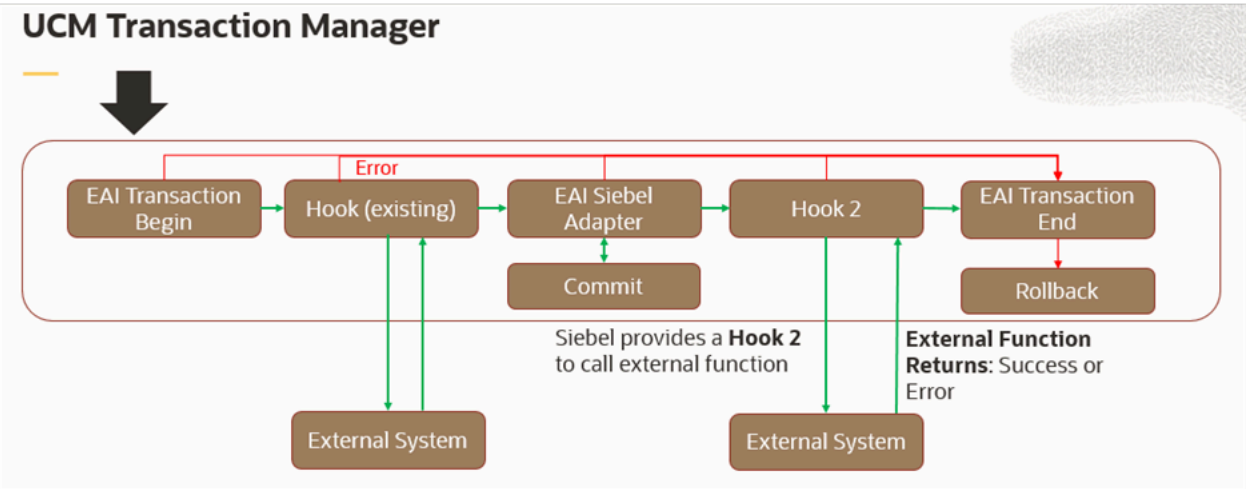
This chapter describes how to configure UCM Transaction Manager's callback functions for Oracle Master Data Management Applications. It includes the following topics:

- *About UCM Transaction Manager Callback Functions*
- *Scenarios for Using the UCM Transaction Manager Callback Functions*
- *Enabling UCM Transaction Manager Callback Functions*
- *Using UCM Utility Service Methods to Simulate External Business Service Callback Functions*
- *Sample eScript for Custom Business Service After Commit*

About UCM Transaction Manager Callback Functions

Oracle Customer Hub (UCM) utilizes UCM Transaction Manager to commit the records as Siebel Database Transaction internally, UCM Transaction Manager invokes EAI Siebel Adapter to perform this task. The callback function is placed before and after EAI Siebel Adapter commits the record. The callback functions are part of the EAI Transaction Block, so if the callback function returns an error all transactions within the transaction block rollback.

The diagram below illustrates UCM Transaction Manager Callback Functions (Hook), inside an EAI Transaction Block:



Scenarios for Using the UCM Transaction Manager Callback Functions

This topic gives the following examples of how UCM Transaction Manager Callback Functions might be use with Oracle Customer Hub (UCM). You might use them differently depending on your business model.

- *Inserting Final Data into the Record before Commit in Siebel*
- *Triggering Synchronous Update in External System*

Inserting Final Data into the Record before Commit in Siebel

If you want to insert some new data, based on the final record, you can use the *before commit* callback function (Hook 1).

In this scenario, Siebel will send the final record, before commit, to external system in the input argument of callback function, whereas the external system modifies the record and sends it back in the output argument of callback function. Siebel then commit this record.

The `SiebelMessage` in the input and output arguments are based on existing UCM Integration Objects, for example: `SwiPersonIO, SwiOrganizationIO, CIFContactUpdate, CIFAccountInsert`, etc.

Following are the sample input and output arguments with Account as parent object and Account Business Address as child entity object:

Sample input & output arguments for hook 1 (before commit) callback function:

```
<Input> //Send to external system
 <SiebelMessage>
 <IntObjectName>SwiOrganizationIO</IntObjectName>
 <ListOfSwiOrganizationIO>
 <ExternalSystemId>Siebel01</ExternalSystemId>
 <Account>
 <MainPhoneNumber>2222222222</MainPhoneNumber>
 <PartyUId>C26C3D88-7C8D-462c-BC5E-35C32358E25F</PartyUId>
 <MainFaxNumber>3333333333</MainFaxNumber>
 <Location>San Jose</Location>
 <Name>TRANSEND</Name>
 <ListOfAccount_BusinessAddress>
 <Account_BusinessAddress>
 <StreetAddress>111 Pine St</StreetAddress>
 <City>San Francisco</City>
 <State>CA</State>
 <Country>USA</Country>
 <PostalCode>94134</PostalCode>
 </Account_BusinessAddress>
 </ListOfAccount_BusinessAddress>
 </Account>
 </ListOfSwiOrganizationIO>
 </SiebelMessage>
</Input>

<Output> //Return from external system
 <SiebelMessage>
 <IntObjectName>SwiOrganizationIO</IntObjectName>
 <ListOfSwiOrganizationIO>
```

```
<ExternalSystemId>Siebel01</ExternalSystemId>
<Account>
<MainPhoneNumber>2222222222</MainPhoneNumber>
<PartyUId>C26C3D88-7C8D-462c-BC5E-35C32358E25F</PartyUId>
<MainFaxNumber>3333333333</MainFaxNumber>
<Location>San Jose</Location>
<Name>TRANSEND</Name>
<AccountTypeCode>Customer</AccountTypeCode> //New data inserted by external system
<AccountStatus>Active</AccountStatus>
<ListOfAccount_BusinessAddress>
<Account_BusinessAddress>
<StreetAddress>111 Pine St</StreetAddress>
<City>San Francisco</City>
<State>CA</State>
<Country>USA</Country>
<PostalCode>94134</PostalCode>
</Account_BusinessAddress>
</ListOfAccount_BusinessAddress>
</Account>
</ListOfSwiOrganizationIO>
</SiebelMessage>
</Output>
```

Triggering Synchronous Update in External System

If you want the data to be in-sync between Siebel and External System, you can use the *after commit* callback function (Hook 2).

In this scenario, the record is first committed successfully in Siebel then Hook 2 invokes to trigger the update in External System. Siebel then sends the before and after record of the EAI Siebel Adapter step in the input argument of the callback function to the External System.

The External System then updates the record in its system. It does not send back the record in the output argument of the callback function, as Siebel commits the record.

In case an error occurs during the update, the External System should send back an error code or error message in the output argument. In such scenario, Siebel rolls back the change in its system and ensures that the data in both systems are in-sync.

Following is the sample input argument with Contact as parent object and Contact Personal Address as child entity object:

Sample input argument for hook 2 (after commit) callback function:

```
<Input> //Send to external system
<SiebelMessage>
<IntObjName>SwiPersonIO</IntObjName>
<SiebelAdapter>Input</SiebelAdapter> //First SiebelMessage is the input to Siebel Adapter
<ListOfSwiPersonIO>
<Contact>
<FirstName>John</FirstName>
<LastName>Doe</LastName>
<EmailAddress>john.doe@gmail.com</EmailAddress>
<WorkPhone>1234567890</WorkPhone>
<ListOfContact_PersonalAddress>
<Contact_PersonalAddress>
<StreetAddress>59 South Grant Avenue</StreetAddress>
<City>San Mateo</City>
<State>CA</State>
<Updated>01/19/1999 02:42:53</Updated>
```

```
<Country>USA</Country>
<PostalCode>94010</PostalCode>
</Contact_PersonalAddress>
</ListOfContact_PersonalAddress>
</Contact>
</ListOfSwiPersonIO>
</SiebelMessage>
<SiebelMessage>
<IntObjName>SwiPersonIO</IntObjName>
<SiebelAdapter>Output</SiebelAdapter> //Second SiebelMessage is the output from Siebel Adapter
<ListOfSwiPersonIO>
<Contact>
<Id>1-Y15</Id> //Got Row Id, since record already commit in Siebel
<FirstName>John</FirstName>
<LastName>Doe</LastName>
<MiddleName></MiddleName>
<IntegrationId></IntegrationId>
<PersonUId>1-Y15</PersonUId>
<PartyUId>85FA3A36-8699-49b2-ADB2-EBCBB02A0EF2</PartyUId>
<Updated>09/08/2022 20:12:10</Updated>
<UpdatedBy>0-1</UpdatedBy>
<operation>update</operation>
<ListOfContact_PersonalAddress>
<Contact_PersonalAddress>
<AddressId>1-826Q</AddressId>
<StreetAddress>59 South Grant Avenue</StreetAddress>
<City>San Mateo</City>
<State>CA</State>
<Updated>06/14/2022 23:59:08</Updated>
<Country>USA</Country>
<PostalCode>94010</PostalCode>
</Contact_PersonalAddress>
</ListOfContact_PersonalAddress>
</Contact>
</ListOfSwiPersonIO>
</SiebelMessage>
</Input>
```

Note: Siebel does not expect any data to be returned in output argument for hook 2 (after commit) callback function, as record has already been committed in Siebel.

In case an error occurs, External System can send back an error code or an error message in the output argument of the callback function (this is true for both hook 1 and hook 2 callback functions).

Following is the format of the error message expected by Siebel:

```
<Output>
 <CustomBSErrorCode>ValidateTest Error</CustomBSErrorCode>
</Output>
```

Enabling UCM Transaction Manager Callback Functions

You can enable the UCM Transaction Manager's callback functions depending on the business model you are using. You can also enable both functions, before commit Hook 1 and after commit Hook 2, by adding new Business Service User Prop using Siebel Tools.

Note: The UCM Transaction Manager's callback functions is not a default function, you must enable it to Configure the UCM Transaction Manager Callback Functions.

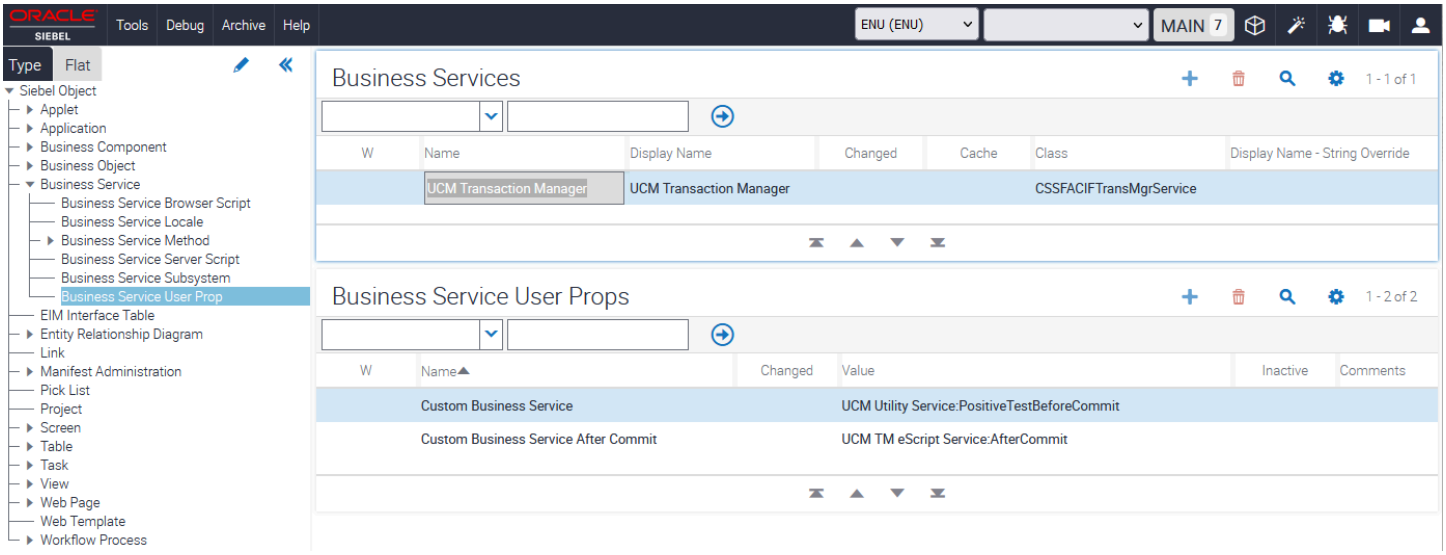
To enable UCM Transaction Manager's callback functions for both real-time and batch process

1. In Siebel Web Tools, create a new workspace to deliver repository changes to runtime database.
2. In Object Explorer, select Business Service: UCM Transaction Manager.
3. Select Business Service User Prop, and create a new record.
4. To enable Hook 1 (before commit), enter the name of new Business Service User Prop: Custom Business Service.

   To enable Hook 2 (after commit), enter the name of new Business Service User Prop: Custom Business Service After Commit.
5. In the value of new Business Service User Prop, enter the Business Service Name and Method, separated by a colon.

   Format: "Business Service Name "+ ": "+ "Business Service Method Name". For example, `UCM Utility Service:PositiveTestAfterCommit`
6. Add checkpoint and submit workspace for delivery.
7. Deliver repository changes to runtime database.

The diagram below shows both callback functions are enable, invoking methods in our internal testing business service *UCM Utility Service* users should create their own custom business service to handle UCM Transaction Manager's callback function.



To enable UCM Transaction Manager's callback functions only for real-time process

1. Create a new workspace to deliver repository changes to runtime database in Siebel Tools.
2. Edit the following workflow processes:

   For CONTACT: UCM Person Customer Profile Integration SOAP Process.

   For ACCOUNT : UCM Organization Customer Profile Integration SOAP Process.
3. Select Workflow Step Transaction Manager and add new input arguments.
4. To specify Business Service Name and Method for hook 1 (before commit), use input argument:

   ```
   Name: "CustomBS", Type: Literal, Value: "UCM Utility Service"
   Name: "CustomMth", Type: Literal, Value: "PositiveTestBeforeCommit"
   ```

5. To specify Business Service Name and Method for hook 2 (after commit), use input argument:

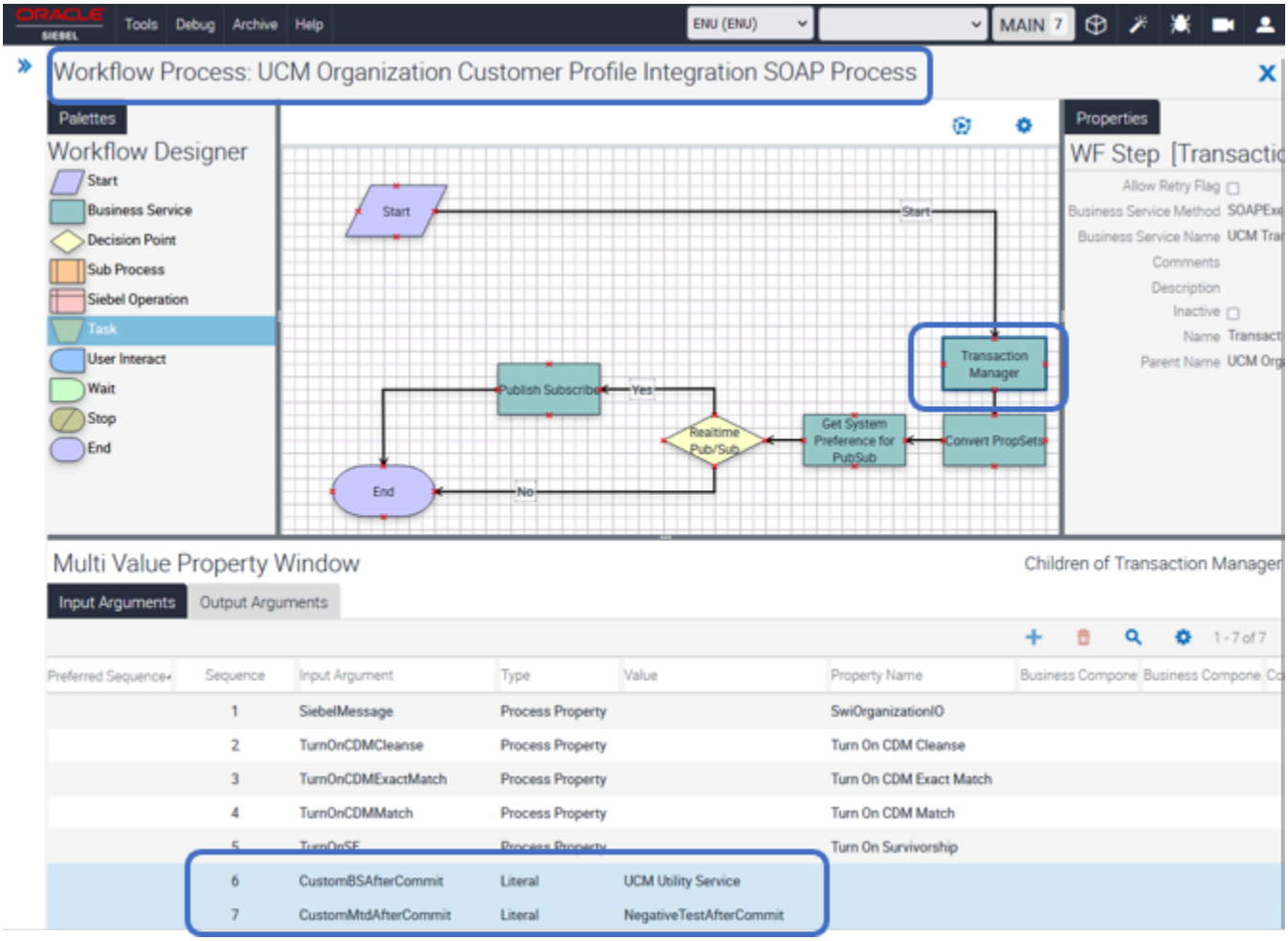
   ```
   Name: "CustomBSAfterCommit", Type: Literal, Value: "UCM Utility Service"
   Name: "CustomMtdAfterCommit", Type: Literal, Value: "NegativeTestAfterCommit"
   ```

6. Add checkpoint and submit workspace for delivery.
7. Deliver repository changes to runtime database.

Note: The Name of the input argument should be exact as above; users should enter their own custom business service and method in the Value.

The diagram below shows a view of real-time SOAP workflow process, when user adds a new input argument for hook 2 (after commit). The value specified in the input arguments overrides the value (if any) specified in UCM Transaction Manager's Business Service User Prop.

For this example, hook 2 will invoke method NegativeTestAfterCommit instead of PositiveTestAfterCommit as specified in Business Service User Prop: Custom Business Service After Commit.



To enable UCM Transaction Manager's callback functions only for batch process

1. Create a new workspace to deliver repository changes to runtime database in Siebel Tools.
2. Edit workflow process: UCM Batch Process.
3. Select workflow Step: Execute Propsets and add new input arguments.

4. To specify Business Service Name and Method for hook 1 (before commit), use input argument:

```
Name: "CustomBS", Type: Literal, Value: "UCM Utility Service"
Name: "CustomMth", Type: Literal, Value: "PositiveTestBeforeCommit"
```

5. To specify Business Service Name and Method for hook 2 (after commit), use input argument:

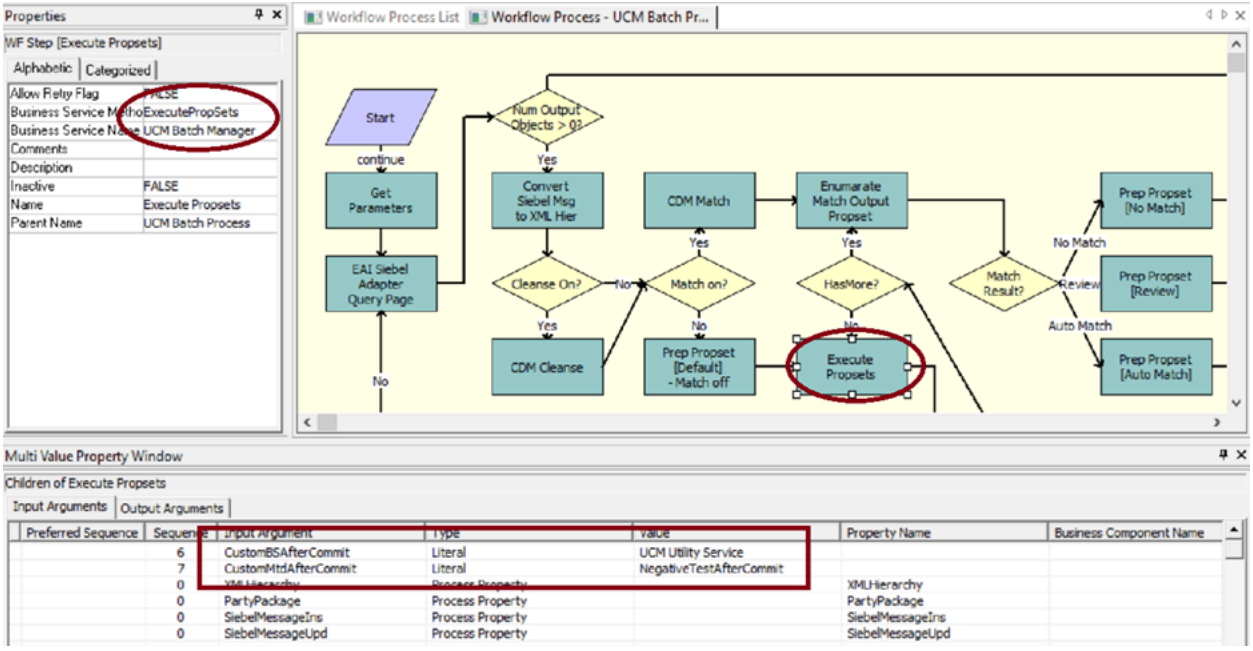
```
Name: "CustomBSAfterCommit", Type: Literal, Value: "UCM Utility Service"
Name: "CustomMtdAfterCommit", Type: Literal, Value: "NegativeTestAfterCommit"
```

6. Add checkpoint and submit workspace for delivery.
7. Deliver repository changes to runtime database.

Note: The Name of the input argument should be exact as above; users should enter their own custom business service and method in the Value.

The diagram below shows a view of UCM Batch Process workflow, when user adds a new input arguments for hook 2 (after commit). The value specified in input arguments overrides the value (if any) specified in UCM Transaction Manager's Business Service User Prop.

For this example, hook 2 will invoke method NegativeTestAfterCommit instead of PositiveTestAfterCommit as specified in Business Service User Prop: Custom Business Service After Commit.



Using UCM Utility Service Methods to Simulate External Business Service Callback Functions

For testing purpose, the UCM Utility Service has additional methods to simulate External Business Service's callback functions:

1. Provide methods to test hook functions internally.

2. Log the function's input and output arguments for validating hook function interfaces.
3. Methods for testing Hook 1 (before commit) callback function:
    - PositiveTestBeforeCommit
    - NegativeTestBeforeCommit
4. Methods for testing Hook 2 (after commit) callback function:
    - PositiveTestAfterCommit
    - NegativeTestAfterCommit

Sample eScript for Custom Business Service After Commit

Once you have configured the user property for Custom Business Service After Commit with a value *CustomerBusinessService:AfterCommit*.

Following is the sample reference for eScript for AfterCommit method.

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)

{

 try

 {

 var returnVal = ContinueOperation;

 if (MethodName =="AfterCommit")

 {

 // Add the logic to process Inputs and call external 3rd party system

 // If (external system update has failed)

 {

 Outputs.SetProperty("CustomBSErrorCode", "Error message from external system.");

 TheApplication().RaiseErrorText("Error message from external system.");

 returnVal = CancelOperation;

 }

 }

 return returnVal;

 }

 catch (e)

 {
```

```
    throw e;

    }

  }
```

12 Configure Siebel UCM using Event Driven Architecture

Configuring Siebel UCM using the Event Driven Architecture

This chapter describes how to configure Siebel UCM using Event Driven Architecture (Siebel CRM Event Pub Sub framework).

- *Overview of Siebel UCM using Event Driven Architecture*
- *Incoming Event Flows*
- *Data Steward Initiated Flows*
- *Configuring Siebel UCM using Event Driven Architecture*

Overview of Siebel UCM using Event Driven Architecture

This feature for Siebel UCM, takes advantage of the Siebel CRM Event Pub Sub framework (released in 23.6) and Kafka Partition Support (released in 24.12) for event driven communications between Siebel CRM and external systems. This allows event publishing to and consuming from Kafka, for various use cases of a typical Siebel UCM implementation. Kafka is a distributed event store and stream-processing platform that is widely used in the industry.

By providing an integration with Kafka for Siebel UCM, this can dramatically simplify the effort and maintenance of having multiple integrations to many customer systems in the organization to process changes to customer master data. Siebel UCM can integrate with many systems via middleware, now we are adding Kafka as another middleware option.

The supported use cases for the Kafka integration are Insert, Update, Delete and Query request events for Account, Contact, Household and Financial Account objects. Submit Merge Request events are also supported for Account and Contact. In addition, Data Steward actions such as Link and Update, Create New, Merge, and Approve events will publish the messages to Kafka.

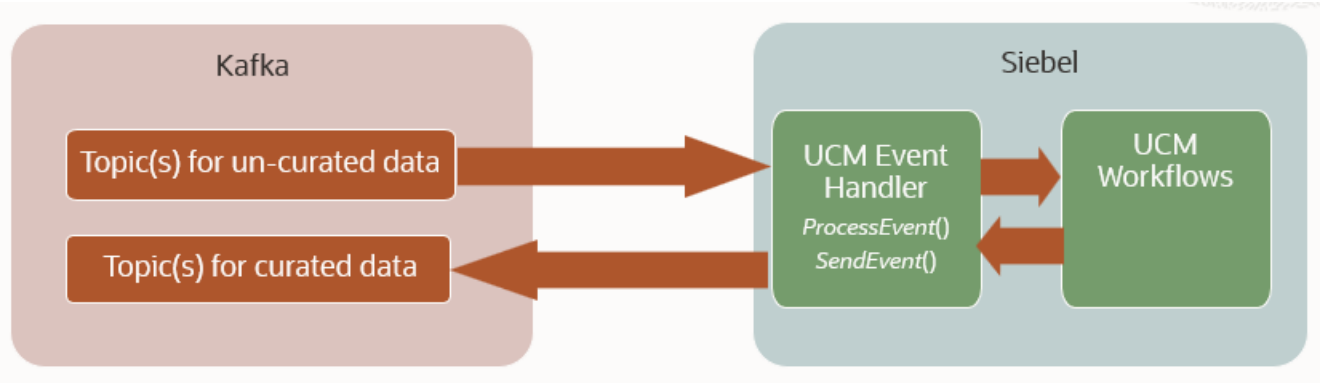
Incoming Event Flows

For an incoming event, the following sequence of information flow takes place:

1. External System will send the un-curated Insert/Update/Delete/Query/SubmitMergeRequest request messages into Kafka topic(s) specific for un-curated request data.
2. UCM subscribes to such Kafka topics(s) and will pick up the request messages for UCM processing.

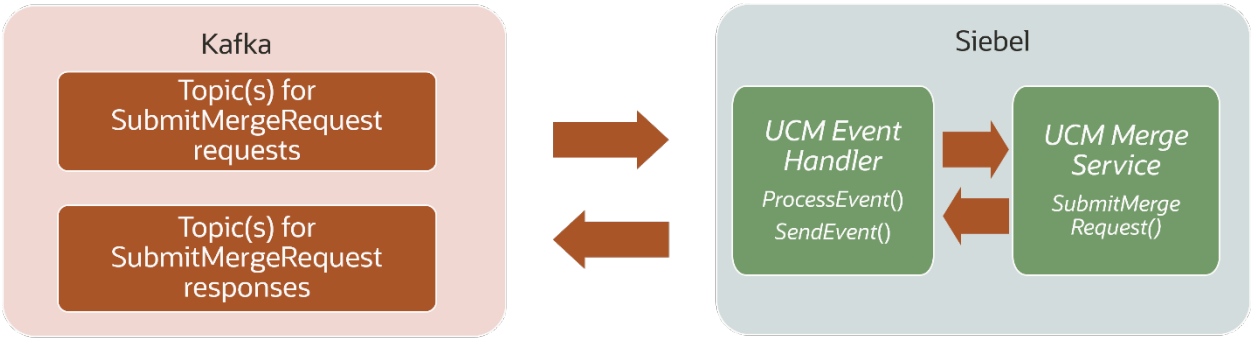
3. UCM will publish the response messages to Kafka topic(s) and optional partition(s) specific for curated response data.

The diagram below illustrates more details of the information flow for Insert/Update/Delete/Query scenarios:



1. UCM subscribes to Kafka Topic(s) for un-curated request data, and request message posted to Kafka Topic triggers business service **UCM Event Handler** method **ProcessEvent**.
2. The business service **UCM Event Handler** method **ProcessEvent** gets the Integration Object name from the request message, looks up the corresponding UCM workflow name in business service **UCM Event Handler** User Properties, invokes the workflow, and get the response message.
3. The business service **UCM Event Handler** method **ProcessEvent** then looks up the Kafka Topic name and optional Partition(s) for curated response data in the business service **UCM Event Handler** User Properties, and invokes business service **Event Handler** method **SendEvent** with Topic name, Partition(s) and response message.
4. The business service **Event Handler** method **SendEvent** publishes the response message to the Kafka Topic and Partition(s).

The diagram below illustrates more details of the information flow for SubmitMergeRequest scenario:

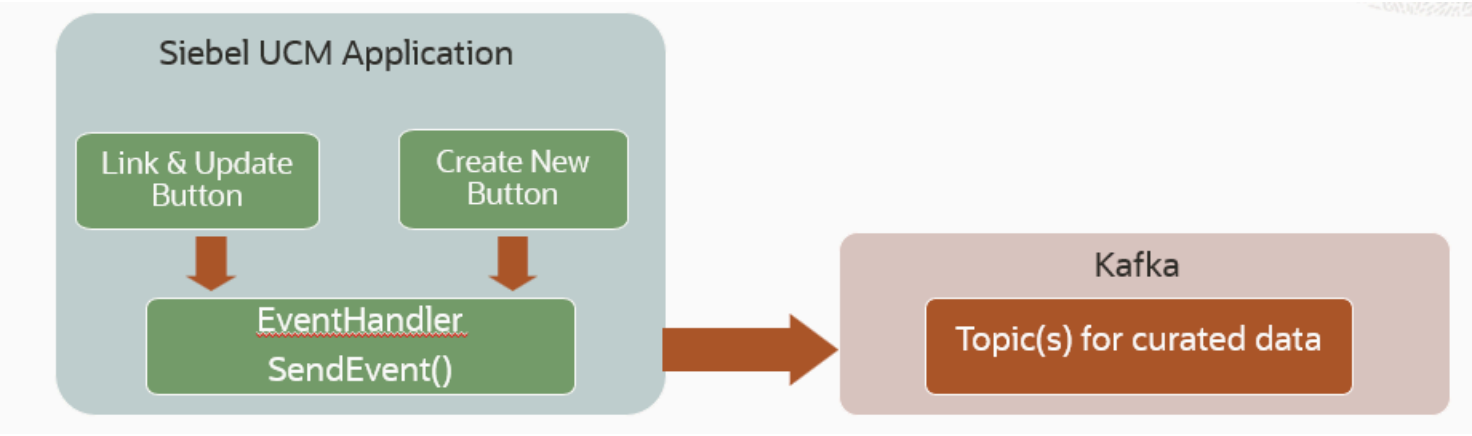


1. UCM subscribes to Kafka Topic(s) for SubmitMergeRequest request data, and request message posted to Kafka Topic triggers business service **UCM Event Handler** method **ProcessEvent**.
2. The business service **UCM Event Handler** method **ProcessEvent** gets the Object Type from the request message, invokes the business service **UCM Merge Service** method **SubmitMergeRequest**, and get the response message.
3. The business service **UCM Event Handler** method **ProcessEvent** then looks up the Kafka Topic name and optional Partition(s) for SubmitMergeRequest response data in the business service **UCM Event Handler** User Properties, and invokes business service **Event Handler** method **SendEvent** with Topic name, Partition(s) and response message.
4. The business service **Event Handler** method **SendEvent** publishes the response message to the Kafka Topic and Partition(s).

Data Steward Initiated Flows

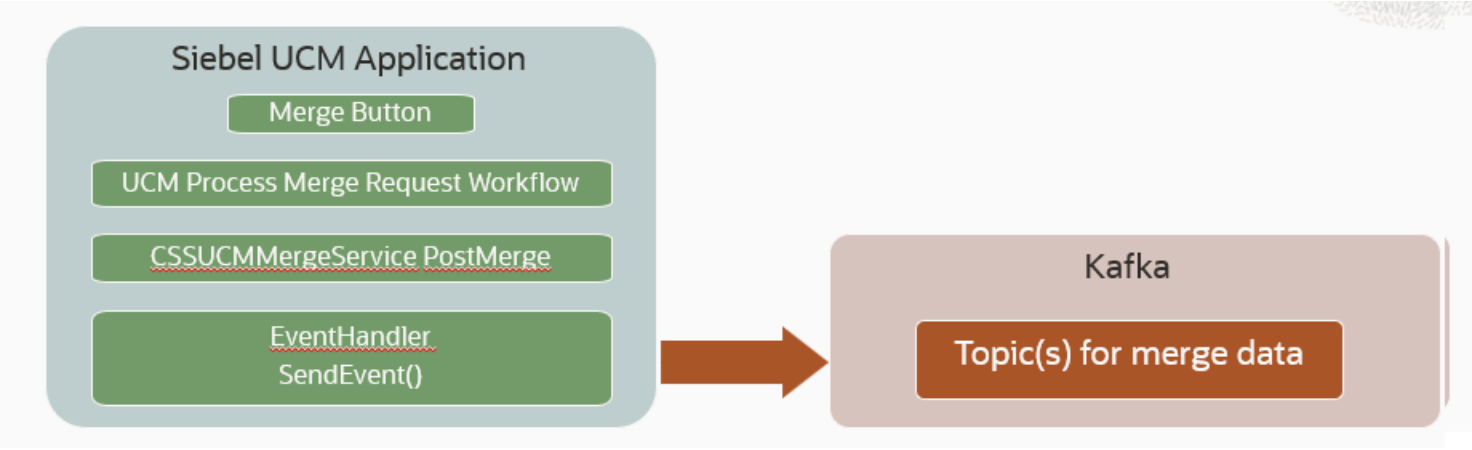
In Link and Update, Create New, Merge, and Approve scenarios, UCM will publish the output messages to the respective Kafka topic(s) and optional partition(s) for curated response data.

The diagram below illustrates the details of the information flow for Link and Update and Create New scenarios:



1. The Link and Update button clicked by Data Steward triggers class **CSSUCMBCDedup** method **Merge**.
2. The Create New button clicked by Data Steward triggers class **CSSUCMBCDedup** method **Promote**.
3. Both methods read the Integration Object name from User Properties in business components **UCM Account DeDup Results** or **UCM Contact DeDup Results,** prepares the instance, invoke business service **UCM Transaction Manager**, look up the Kafka Topic names and optional partition(s) for curated response data in the business service **UCM Event Handler** User Properties, and invoke business service **Event Handler** method **SendEvent** with Topic name , Partition(s) and output message.
4. The business service **Event Handler** method **SendEvent** publishes the output message to the Kafka Topic and Partition(s).

The diagram below illustrates more details of the information flow for Merge scenario:

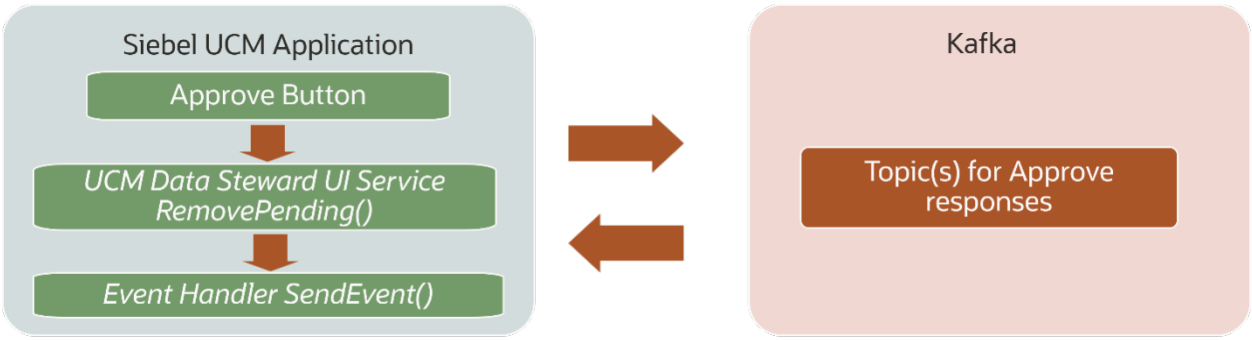


1. The Merge button clicked by Data Steward triggers the workflow **UCM Process Merge Request Workflow**.
2. Near the end of workflow, business service **UCM Merge Service** method **PostMerge** is invoked.
3. At the end of the method, it reads the Publish IO name from business service **UCM Merge Service** User Properties, prepares the instance, looks up the Kafka Topic name and optional Partition(s) for curated response

data in the business service **UCM Event Handler** User Properties, and invokes business service **Event Handler** method **SendEvent** with Topic name, Partition(s) and output message.

4. The business service **Event Handler** method **SendEvent** publishes the output message to the Kafka Topic and Partition(s).

The diagram below illustrates more details of the information flow for Approve scenario:



1. The Approve button clicked by Data Steward triggers the business service **UCM Data Steward UI Service** method **RemovePending**.
2. At the end of the method, it looks up the Kafka Topic name and optional Partition(s) for Approve response data in the business service **UCM Event Handler** User Properties, and invokes business service **Event Handler** method **SendEvent** with Topic name, Partition(s) and output message.
3. The business service **Event Handler** method **SendEvent** publishes the output message to the Kafka Topic and Partition(s).

Configuring Siebel UCM using Event Driven Architecture

This topic describes how to configure Siebel UCM using Event Driven Architecture.

For background information on configuring Siebel CRM Event Publication and Subscription, see *Siebel System Administration Guide*.

The following System Preference value needs to be set to TRUE to enable this feature:

System Preference	Default Value	Description
Enable UCM using EDA	FALSE	Enables Siebel UCM using Event Driven Architecture

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

For un-curated request messages, use topic name **ExternalSystemToUCM** when creating topic in Kafka. Refer to Kafka documentation for more information on creating topics. Otherwise, if a customized topic name is used, the following configuration change needs to be done:

1. Go to **Sitemap → Administration-Application**.
2. Click **Event Subscription**.

3. Search for the following entry:

Service	Method	Event Store
UCM Event Handler	ProcessEvent	ExternalSystemToUCM

4. Change the value for field Event to the customized Kafka topic name.

For curated response messages, use topic name **UCMToExternalSystem** when creating topic in Kafka. Refer to Kafka documentation for more information on creating topics. Otherwise, if customized topic name(s) and/or partition(s) are used, the values of the following user properties in business service **UCM Event Handler** needs to be changed to the customized Kafka topic name(s) and partition(s) in the format of `"<Topic>;<Partitions>"` where `"<Topic>"` is required and `";<Partitions>"` is optional, for example, `"UCMToExternalSystem;0,1,3"`:

User Property	Default Value	Description
UCM Workflow Response For SwiOrganizationIO	UCMToExternalSystem	Topic Name and optional Partition(s) for curated Insert/Update/Delete response messages for integration object **SwiOrganizationIO**
UCM Workflow Response For SwiPersonIO	UCMToExternalSystem	Topic Name and optional Partition(s) for curated Insert/Update/Delete response messages for integration object **SwiPersonIO**
UCM Workflow Response For SwiGroupIO	UCMToExternalSystem	Topic Name and optional Partition(s) for curated Insert/Update/Delete response messages for integration object **SwiGroupIO**
UCM Workflow Response For SwiFinancialAssetIO	UCMToExternalSystem	Topic Name and optional Partition(s) for curated Insert/Update/Delete response messages for integration object **SwiFinancialAssetIO**
UCM Query Workflow Response For SwiOrganizationIO	UCMToExternalSystem	Topic Name and optional Partition(s) for Query response messages for integration object **SwiOrganizationIO**
UCM Query Workflow Response For SwiPersonIO	UCMToExternalSystem	Topic Name and optional Partition(s) for Query response messages for integration object **SwiPersonIO**
UCM Query Workflow Response For SwiGroupIO	UCMToExternalSystem	Topic Name and optional Partition(s) for Query response messages for integration object **SwiGroupIO**
UCM Query Workflow Response For SwiFinancialAssetIO	UCMToExternalSystem	Topic Name and optional Partition(s) for Query response messages for integration object **SwiFinancialAssetIO**
UCM Merge Request Response For Organization	UCMToExternalSystem	Topic Name and optional Partition(s) for SubmitMergeRequest response messages for Organization

User Property	Default Value	Description
UCM Merge Request Response For Person	UCMToExternalSystem	Topic Name and optional Partition(s) for SubmitMergeRequest response messages for Person
UCM Link And Update For SwiOrganizationIO	UCMToExternalSystem	Topic Name and optional Partition(s) for curated Link and Update output messages for integration object **SwiOrganizationIO**
UCM Link And Update For SwiPersonIO	UCMToExternalSystem	Topic Name and optional Partition(s) for curated Link and Update output messages for integration object **SwiPersonIO**
UCM Create New For SwiOrganizationIO	UCMToExternalSystem	Topic Name and optional Partition(s) for curated Create New output messages for integration object **SwiOrganizationIO**
UCM Create New For SwiPersonIO	UCMToExternalSystem	Topic Name and optional Partition(s) for curated Create New output messages for integration object **SwiPersonIO**
UCM Merge Response For SwiOrganizationPublishIO	UCMToExternalSystem	Topic Name for curated Merge output messages for integration object **SwiOrganizationPublishIO**
UCM Merge Response For SwiPersonPublishIO	UCMToExternalSystem	Topic Name and optional Partition(s) for curated Merge output messages for integration object **SwiPersonPublishIO**
UCM Approve Response For SwiOrganizationPublishIO	UCMToExternalSystem	Topic Name and optional Partition(s) for curated Approve output messages for integration object **SwiOrganizationPublishIO**
UCM Approve Response For SwiPersonPublishIO	UCMToExternalSystem	Topic Name and optional Partition(s) for curated Approve output messages for integration object **SwiPersonPublishIO**

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

13 Using Oracle Data Goverance Manager

Using Oracle Data Goverance Manage

This chapter contains information on Oracle Data Governance Manager. It includes the following topics:

- *About Oracle Data Governance Manager*
- *Overview of Data Governance Manager Modules*
- *Scenarios for Using Oracle Data Governance Manager*
- *Configuring Cleanse Functionality*
- *Viewing Mastered Records*
- *Administering Batch Imports*
- *Viewing Incomplete Records*
- *Viewing Merge Activity*
- *Viewing the Suspect Queue and Performing a Merge*

About Oracle Data Governance Manager

Oracle Data Governance Manager is a tool that works with Oracle Customer Hub (UCM) to allow a data steward to manage data through five major functional modules. It helps the data steward operate and monitor the different Oracle Master Data Management Applications functions, such as data consolidation, master record creation, data cleansing, and data sharing between subscribing applications.

> **Note:** Oracle Data Governance Manager runs with Oracle WebLogic server and includes a restricted use license for all Oracle Data Governance Manager related implementations. For more information, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

For details of the Web services used to perform operations for Oracle Data Governance Manager modules, see *Oracle Customer Hub (UCM) and Data Governance Manager Web Services Reference*.

Logging into Oracle Data Governance Manager

The following procedure describes how to launch Oracle Data Governance Manager.

To log in to Oracle Data Governance Manager

1. Start Oracle Data Governance Manager.
2. Select the correct language from the Language list.
3. Enter your username and password.
4. In the From and To fields, select the date range for the data that you want to view.
5. In the Hub list, select UCM.
6. Click Login.

Overview of Data Governance Manager Modules

Oracle Data Governance Manager has the following modules to manage the data lifecycle:

- *Master Module*
- *Consolidate Module*
- *Cleanse Module*
- *Govern Module*
- *Share Module*

Master Module

The Master module displays the number of records that has been created and or updated within a configurable period. The Master screen displays the results for Contact records and Account records, or both. Data stewards can view details of individual accounts or contacts. For information on using the Master module, see *Viewing Mastered Records*.

Consolidate Module

The Consolidate module enables the data steward to manage batch import activities. From this screen, the data steward can start data batch jobs, view completed and pending jobs, and perform corrections on any records rejected from a batch job. For information on using the Consolidate module, see *Administering Batch Imports*.

Cleanse Module

The Cleanse module enables the data steward to continually monitor data quality through configurable metrics. The screen also displays registered external systems to allow data stewards to identify problems. For information on using the Cleanse module, see *Viewing Incomplete Records*.

Govern Module

The Govern module enables data stewards to track and define enterprise-wide definitions of data, metadata, and rules that dictate how the enterprise is run. Monitoring and fixing functions allow the data steward to enforce the rules. Additionally, from this module, the data steward can set up links to reference documents that contain data contract and definitions within the enterprise. For information on using the Govern module, see *Viewing the Suspect Queue and Performing a Merge*.

Share Module

The Share module allows data stewards to view Oracle Customer Hub (UCM) activity in real-time. They can view updates, merge and unmerge operations. Additionally, they can track how many publish operations are being sent to subscribing systems and how many queries are being requested from Oracle Customer Hub (UCM). For information on using the Share module, see *Viewing Merge Activity*.

Scenarios for Using Oracle Data Governance Manager

This topic describes how Oracle Data Governance Manager might be used. You might use Oracle Data Governance Manager differently, depending on your business model. It includes the following scenarios:

- *Using Data Governance Home Page*
- *Using Data Governance Master Data Module (Data Steward)*
- *Consolidating Records in Oracle Data Governance Manager*
- *Viewing and Fixing Rejected Records in Oracle Data Governance Manager*
- *Viewing and Fixing Multiple Rejected Records in Oracle Data Governance Manager*
- *Consolidating Completed Records in Oracle Data Governance Manager (Data Steward)*
- *Manually Reviewing Suspect Records in Oracle Data Governance Manager (Data Steward)*
- *Manually Reviewing Completed Records in Oracle Data Governance Manager*
- *Viewing Graphs of All Records That Were Part of Batch Jobs*
- *Cleansing Records in Data Governance Manager*

Using Data Governance Home Page

A data steward logs in to Oracle Data Governance Manager and views the home page. From the home page, the data steward can choose from five modules: Master, Govern, Share, Cleanse, and Consolidate. The data steward drills down on the details of the five modules to perform data governance tasks. A login screen validates the user ID and password. Data access is determined by the views already set up in Oracle Customer Hub (UCM).

Using Data Governance Master Data Module (Data Steward)

From the home page, the data steward clicks the Master section to view the activity records on mastered records in Oracle Customer Hub (UCM). The Master Data section displays a summary of the mastered account and contact records for each day, week, month, and all options, which query against all data in Oracle Customer Hub (UCM) Server. The results are displayed graphically.

Consolidating Records in Oracle Data Governance Manager

From the home page, the data steward clicks the Consolidate section. From the Consolidate screen the data steward can view all batch jobs with the related statistics run within a defined period, such as the past hour, day, week, year, and so on.

Viewing and Fixing Rejected Records in Oracle Data Governance Manager

From the home page, the data steward clicks the Consolidate section. From the Consolidate screen the data steward clicks the Rejected Records pie chart from a batch report to view all rejected records from that batch job. If the data steward finds a record that was rejected by mistake, corrections can be made from this view, and the fixed records are saved when the batch job is rerun. Additionally, the data steward might update multiple batch records at once.

Viewing and Fixing Multiple Rejected Records in Oracle Data Governance Manager

From the home page, the data steward clicks the Consolidate section. From the Consolidate screen the data steward clicks the Rejected Records pie chart from a batch report to view all rejected records from that batch job. If the data steward finds a record that was rejected by mistake, corrections can be made from this view, and the fixed records are saved when the batch job is rerun. Additionally, the data steward might update multiple batch records at once.

Consolidating Completed Records in Oracle Data Governance Manager (Data Steward)

From the home page, the data steward clicks the Consolidate section and views records processed by a batch job. The data steward can view the total number of inserted and updated records from the batch process.

Manually Reviewing Suspect Records in Oracle Data Governance Manager (Data Steward)

The Suspect Record Details allows the data steward to choose whether records in the queue are duplicates. From this screen, the data steward can either merge two duplicate records or create a new record. The data steward can then approve or merge the records. *To approve* means that the two suspect records are processed as separate records. *To merge* means the data steward considers the two records to be the same and merges them during the next batch process.

Manually Reviewing Completed Records in Oracle Data Governance Manager

Following a manual merge process the Batch Import screen is displayed. From this screen, the data steward can read the flat file structures and start the batch import processes. Oracle Customer Hub (UCM) Web services enable batch template files to be pulled from Oracle Customer Hub (UCM). After the data steward chooses the template choice, the data steward can view the content of the subsequent data file before importing the data in to Oracle Customer Hub (UCM). Following review of the data, the data steward clicks the Import button. Any errors during the file parsing are captured in an error table that can then be reviewed and acted upon by the data steward.

Viewing Graphs of All Records That Were Part of Batch Jobs

A data steward logs in to Oracle Data Governance Manager and views the home page. From the home page, the data steward chooses the Share module. The Share screen allows the data steward to view a graphical representation of all records that were part of batch jobs using the following criteria: modified, unmerge, and merge.

Cleansing Records in Data Governance Manager

From the Data Quality screen, the data steward can monitor the integrity of Oracle Customer Hub (UCM) data through the Completeness metric. Data completeness is measured at both the master records level and at the source system level. The data steward can click each node from the system list to view details about the incomplete data from each node.

Configuring Cleanse Functionality

You set different metrics for the cleanse functionality in the Cleanse module. Use the following task to configure cleanse functionality for the source completeness metric.

To configure the cleanse functionality for the source completeness metric

1. In Oracle Data Governance Manager, click Cleanse.
2. To change the source completeness metric, click the Source completeness list.
3. Select the fields that you want to include as part of the source completeness metric, which are run against the Source Data History tables.
4. Click Refresh.

   The source systems connected to Oracle Customer Hub (UCM) using spokes are updated.

Use the following task to configure cleanse functionality for the Oracle Customer Hub completeness metric.

To configure the cleanse functionality for the Oracle Customer Hub completeness metric

1. In Oracle Data Governance Manager, click Cleanse.
2. Click the Hub Completeness list to change the Hub Completeness metric.
3. Select the fields that you want to include as part of hub completeness metric, which is run against the Best version tables.
4. Click Refresh.

   The systems connected to Oracle Customer Hub (UCM) are updated.

Viewing Mastered Records

The Master module enables you to view the details of mastered account and contact records. *Mastered records* are records that were created or updated. You can limit the view of the mastered records by setting the period and other constraints. Use the following task to view mastered records.

To view mastered records

1. From the Oracle Data Governance Manager home page, click Master.

   The Master Data view appears. A pie chart and bar graph provide an illustration of the mastered records, displaying the Contact records and Account records.
2. Click either graph for more information on the mastered records.

   The Mastered Accounts or Mastered Contacts screen appears, displaying a detailed view of each mastered record.
3. Click the View menu to choose the columns to be displayed.
4. Click the Object Type list to choose which object type to display: contact or account.
5. Enter a date range for the data displayed in the From and To fields.
6. Click the Quick Query list to select the query values, such as Today, Last Week, then click Refresh for a broader summary of the records.

Administering Batch Imports

The Consolidate module enables you to manage import operations in Oracle Customer Hub (UCM). Besides running new batch jobs using prebuilt template files, you can do the following:

- Report on batch imports completed or currently running.
- View batch reports grouped by completed rejected criteria
- Fix any rejected records and then restart a batch process.
- View how completed records were handled by the batch process, such as records that were inserted or updated or both.

Use the following task to run a batch import job.

To run a batch import job

1. From the Oracle Data Governance Manager home page, click Consolidate.
2. In the Consolidate screen, click the File Import tab.
3. From the Choose Template list, select a mapping template.
4. From the Delimiter list, choose a delimiter, if necessary.
5. Click Browse, then select a batch file.
6. Click Load.

   The file information appears in the window.
7. Click Import.

   The records from the flat files contained in the batch job to the UCM SDH staging tables. When the loading is complete, the batch workflow is started, and the records are moved from SDH tables and saved to the base tables.

Use the following task to view a batch job.

To view a batch job

1. After successfully loading a batch file and initiating the import in Oracle Customer Hub (UCM), click the Batch Report tab.

   The batch job that is currently running is displayed. The Completed, Pending, and Jobs in Progress are also displayed.
2. Click any part of the pie chart to view information about the batch jobs.

Use the following task to view rejected records from a batch job.

To view rejected records from a batch job

1. From the Consolidate screen Batch Report view, click the Rejected segment of the pie chart.

   The Rejected Batch Details view appears.
2. In the By Object Type, choose Account or Contact from the list.
3. In the Error Message and Error Code field, view details about the record was rejected for inclusion into the UCM SDH tables.
4. Make the required changes to the record, then click Save.

   The record will be queued into the next batch job.

Use the following task to view completed batch records.

To view completed batch records

1. From the Consolidate screen Batch Report view, click the Completed segment of the pie chart.

   The Completed Batch Details view appears.
2. In the By Object Type, choose Account or Contact from the list.
3. View a graph of the insert and update operations made to the chosen object.

Viewing Incomplete Records

The Cleanse module enables you to view the ratio of completed records in your external systems and in Oracle Customer Hub (UCM). Data quality is measured by its completeness definitions. Oracle Data Governance Manager uses the completeness definitions to control data quality in Oracle Customer Hub (UCM). Use the following task to view incomplete records.

To view incomplete records

1. From the Oracle Data Governance Manager home page, click Cleanse.

   The Data Quality screen appears displaying Oracle Customer Hub (UCM) and all external systems to which it is connected.
2. Move the pointer over any record for a brief view of complete and incomplete records.
3. Click the Isolate button on any record to view the system record in an isolated view.

   To return to the full Data Quality view, click the Isolate button again.
4. Click the Object Type list to choose between Contact and Account records.
5. Use the Hub Completeness Fields and Source Completeness Fields lists to determine the completeness of definitions.

   The definitions include All, Name, Location, and so on.
6. Use the From and To fields to determine the date or time range in which to view records.
7. Use the Click the Quick Query list to select the query values, such as Today, Last Week, then click Refresh for a broader summary of records.

Viewing Merge Activity

The Share module enables you to track the number of merges and the period during which merges were executed. This module enables you to view the number of modified, merged, and unmerged records entering Oracle Customer Hub (UCM). Use the following task to view merge activity in the Share screen.

To view merge activity in the Share screen

1. From the Oracle Data Governance Manager home page, click Share.
2. In the Share screen, click the Account or Contact tab to view the record type.
3. View a bar graph representing the merged, modified, and unmerged records.
4. Double click any group to view a detailed list of the merged, modified, or unmerged records.

Viewing the Suspect Queue and Performing a Merge

The Govern module enables you to start different parts of Oracle Customer Hub (UCM) from Oracle Data Governance Manager. Additionally, you might view the suspect queue for suspect records. The suspect queue enables you to view

a queue of all suspect records for a given period. You might choose records from potential matches and then perform a merge. Use the following task to perform a merge operation from the Suspect queue.

To perform a merge from the Suspect queue

1. From the Oracle Data Governance Manager home page, click Govern.

   The Governance screen appears.
2. Click the Suspect Queue link.
3. From the Object Type list, choose Account or Contact.

   The suspect records from the chosen object are displayed in the Suspect Queue screen, and the Potential Match view displays any potential matches already in Oracle Customer Hub (UCM).
4. In the Potential Match view, select the match record.
5. Click the Merge button to merge the two records, then click Approve to commit the record to the SDH table.

   A window appears indicating success following the merge operation.
6. Click OK.

14 Configuring the MDM Foundation and Workflows

Configuring the MDM Foundation and Workflows

This chapter contains information on workflows used with Oracle Customer Hub (UCM) and configuration tasks for the Oracle Master Data Management Foundation. It includes the following topics:

- *Configuring Workflows and the MDM Foundation*
- *Process of Configuring Inbound Communication Using Siebel EAI MQSeries Transport*
- *Process of Configuring Siebel EAI JMS Transport to Integrate Data*
- *Process of Configuring Siebel EAI HTTP Transport*
- *About the Integration of Oracle Customer Hub (UCM) with Siebel Business Applications*
- *Extending the Integration of Oracle Customer Hub (UCM) with Siebel Business Applications*
- *Configuring Siebel Master Data Integration Objects*
- *Cleansing Batch Data and Matching Data*

Configuring Workflows and the MDM Foundation

The following topics describe how to set up workflows and configure the MDM Foundation:

- *About Oracle Master Data Management Applications Integration Services*
- *Configuring UCM Server Components for Data Management and Survivorship*
- *Configuring Run-Time Events for UI Interaction*
- *Configuring Default System for UI Interaction*
- *Configuring UCM Inbound Server Communication*
- *Configuring MQSeries Server Receiver of Siebel Server Component*
- *Configuring JMS Receiver Siebel Server Component*
- *Configuring Oracle Customer Hub (UCM) to Run the Extension Service*
- *About the Integration of Oracle Customer Hub (UCM) with Siebel Business Applications*
- *Extending the Integration of Oracle Customer Hub (UCM) with Siebel Business Applications*
- *Configuring Siebel Master Data Integration Objects*
- *About Siebel Master Data Application Integration Objects*
- *Modifying Siebel Master Data Application Integration Objects*
- *Extending UCM Integration Objects*
- *Adding the Status Key Property to Oracle Customer Hub (UCM) Integration Objects*
- *About the Exact Match Functionality*

- *Configuring Exact Match Functionality in Oracle Customer Hub (UCM)*
- *Configuring Siebel Data Quality Cleansing for Oracle Customer Hub (UCM)*
- *Configuring Siebel Data Quality Cleansing for Oracle Customer Hub (UCM)*

About Oracle Master Data Management Applications Integration Services

Oracle Master Data Management Applications provide objects called *business services*, which you can reuse in multiple applications. These business services perform the insert, update, query, and delete operations on data in Oracle Master Data Management Applications. 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 Business Service*
- *UCM Cross Reference Business Service*
- *UCM Dispatcher Business Service*
- *UCM Security Manager Business Service*
- *UCM Data Quality Manager*
- *UCM Batch Manager Business Service*
- *UCM Survivorship Engine*
- *UCM Account Source Data and History Service*
- *UCM Contact Source Data and History Service*
- *UCM FINCORP Account Source Data and History Service*
- *UCM Household Source Data and History 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, you must 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 more information on the integration services and Oracle Master Data Applications architecture, see *About Oracle Customer Hub (UCM) Architecture* and other topics in *Architecture and Framework of Oracle Customer Hub (UCM)*.

UCM Transaction Manager Business Service

You can extend this business service by manipulating its Operational user property. The following information lists the user properties examples for the UCM Transaction Manager. Use the following basic format to enter values for each operation.

CIFOperation_Query

CIFOperation_Query user properties use the following format:

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

where:

- Service, Method, and Argument are separated by a slash mark (/).
- Each argument ends with a semicolon (;).

CIFOperation_XMLQuery

CIFOperation_XMLQuery user properties use the following format:

```
EAI Siebel Adapter/Query/#XMLHierarchy;
```

where:

- 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

CIFOperation_GetValue user properties use the following format:

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

where:

- *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.

The following table lists the user properties examples for the UCM Transaction Manager.

UCM Transaction Manager User Property Name	Value
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

UCM Transaction Manager User Property Name	Value
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
DisableSessionRealtimeClns	TRUE
DisableSessionRealtimeDeDup	TRUE
DispatcherMapName	CIFDispMap
Enable Updating SDH Type on Error	TRUE
EnableAgentLock	TRUE
EnableCrossReferenceForQuery	None
Household Address Field 1	Street Address;Street Address
Household Address Field 2	City;City
Household Address Field 3	State;State
Household Address Field 4	Postal Code;Postal Code
Household Address Field 5	Country;Country
Household Address Primary CUT Address	Primary Household Address Id;Household_Household Address
Insert_IOandOp_Account	CIFAccountInsert;CIFAccountInsertRs;IXMLOperation_ADD
Insert_IOandOp_Contact	CIFContactInsert;CIFContactInsertRs;IXMLOperation_ADD
Insert_IOandOp_FINCORP Account	CIFFINCORPAccountInsert;CIFFINCORPAccountInsertRs;IXMLOperation_ADD
Insert_IOandOp_Household	CIFHouseholdInsert;CIFHouseholdInsertRs;IXMLOperation_ADD
IONameForQueryBV_Account	CIFAccount

UCM Transaction Manager User Property Name	Value
IONameForQueryBV_Contact	CIFContact
IONameForQueryBV_FINCORP Account	CIFFINCORPAccount
IONameForQueryBV_Household	CIFHousehold
IXMLOperation_ADD	EAI Siebel Adapter/Insert/
IXMLOperation_DELETE	EAI Siebel Adapter/Delete/
IXMLOperation_QUERY	EAI Siebel Adapter/Query/
IXMLOperation_QUERY_BY_ID	EAI Siebel Adapter/Query/PrimaryRowId;!SiebelMessage;
IXMLOperation_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/
MaxNumQueryOutputObjects	50
Operation Delete	IXMLOperation_DELETE
Operation Insert	IXMLOperation_ADD
Operation Query	IXMLOperation_QUERY;IXMLOperation_QUERY_BY_ID;SOAPOperation_Query
Operation Update	IXMLOperation_UPDATE
Operation Upsert	IXMLOperation_UPSERT
Primary Object 1	Address/CIFPersonalAddress
RealtimeInsertAutoMatch_Account	CIFAccountUpdate;CIFAccountUpdateRs;IXMLOperation_UPDATE
RealtimeInsertAutoMatch_Contact	CIFContactUpdate;CIFContactUpdateRs;IXMLOperation_UPDATE

UCM Transaction Manager User Property Name	Value
RealtimeInsertAutoMatch_Household	CIFHouseholdUpdate;CIFHouseholdUpdateRs;IXMLOperation_UPDATE
SDHCommitSize	5
SOAPOperation_Delete	EAI Siebel Adapter/Delete/
SOAPOperation_Insert	EAI Siebel Adapter/Insert/
SOAPOperation_Query	EAI Siebel Adapter/Query/
SOAPOperation_Update	EAI Siebel Adapter/Update/
SOAPOperation_Upsert	EAI Siebel Adapter/Upsert/
Update_IOandOp_Account	CIFAccountUpdate;CIFAccountUpdateRs;IXMLOperation_UPDATE
Update_IOandOp_Contact	CIFContactUpdate;CIFContactUpdateRs;IXMLOperation_UPDATE
Update_IOandOp_FINCORP Account	CIFFINCORPAccountUpdate;CIFFINCORPAccountUpdateRs;IXMLOperation_UPDATE
Update_IOandOp_Household	CIFHouseholdUpdate;CIFHouseholdUpdateRs;IXMLOperation_UPDATE
Upsert_IOandOp_Account	CIFAccountUpsert;CIFAccountUpsertRs;IXMLOperation_UPSERT
Upsert_IOandOp_Contact	CIFContactUpsert;CIFContactUpsertRs;IXMLOperation_UPSERT
Upsert_IOandOp_FINCORP Account	CIFFINCORPAccountUpsert;CIFFINCORPAccountUpsertRs;IXMLOperation_UPSERT
Upsert_IOandOp_Household	CIFHouseholdUpsert;CIFHouseholdUpsertRs;IXMLOperation_UPSERT

UCM Cross Reference Business Service

The user properties for this business service appear in the following table.

UCM Cross Reference Property Names	Value
CompFieldMap Account_Business Address	CIF Account Address Reference:Address Id;;Address Id
CompFieldMap Contact_Alternate Phone	CIF Alternate Phone Reference:Alternate Phone Id;;Alternate Phone Id

UCM Cross Reference Property Names	Value
CompFieldMap Contact_Communication Address	CIF Communication Address Reference:Communication Address Id;;Communication Address Id
CompFieldMap Contact_INS Personal Address	CIF Contact Address Reference:Address Id;;INS Personal Address Id
CompFieldMap FINCORP Account	CIF FINCORP Account Reference:FINCORP Account Id;;Id
CompFieldMap FINCORP Deposit Account	CIF FINCORP Deposit Account Reference:FINCORP Deposit Account Id;;Id
CompFieldMap FINCORP Loan Account	CIF FINCORP Loan Account Reference:FINCORP Loan Account Id;;Id
CompFieldMap UCM HE Constituent Name	CIF Contact Name Reference:Contact Name Id;;Id
EnableChildCrossReferenceForQuery	None
Generate UID Service Method	GenerateID
Generate UID Service Method Arg 1	IDType:GUID
Generate UID Service Name	FINS Teller Converter Extensions
MaxNumChildCrossReferenceForQuery	10

UCM Dispatcher Business Service

You can modify both user properties for this business service as shown in the following table.

UCM Dispatcher User Property Name	Value
DispatcherMapName	The name of the dispatcher map shipped or that you customized.
XMLEnvIntObjectName	The name of the envelope integration object shipped or that you customized.

UCM Security Manager Business Service

You can modify or extend this business service by manipulating its user properties. The following table displays these user properties.

UCM Security Manager User Property Name	Value
IXMLOperation_ADD	Insert
IXMLOperation_DELETE	Delete
IXMLOperation_QUERY	Query
IXMLOperation_QUERYPAGE	Query
IXMLOperation_UPDATE	Update
IXMLOperation_UPSERT	Upsert

UCM Data Quality Manager

The user properties for this business service appear in the following table.

UCM Data Quality Manager User Property Name	Value
Account Account Cleansing Field 1	Name:Name;Location:Location;
Account Account Matching Field 1	Name:Name;DUNS Number:DUNS Number;
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:Primary Account Street Address;City:Primary Account City;State:Primary Account State;Postal Code:Primary Account Postal Code;Country:Primary Account 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;

UCM Data Quality Manager User Property Name	Value
Contact Contact Matching Field 1	Last Name:Last Name;First Name:First Name;Middle Name:Middle Name;Work Phone #:Work Phone #;Cellular Phone #:Cellular Phone #;Home Phone #:Home Phone #;
Contact Contact Matching Field 2	Email Address:Email Address;Social Security Number:Social Security Number;Birth Date:Birth Date;
Contact Contact_Account Cleansing Field 1	Account:Name;Account Location:Location;
Contact Contact_Account Matching Field 1	Account:Account;
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:Primary Personal Street Address;INS Personal City:Primary Personal City;INS Personal State:Primary Personal State;INS Personal Postal Code:Primary Personal Postal Code;INS Personal Country:Primary Personal Country;
Contact Manual Threshold	70
Contact Matching Component 1	Contact:Contact;Primary Contact_INS Personal Address;Contact_Account;
Enable XRef Match	false
ExactMatch Object 1	Account:CIFAccount;Contact:CIFContact;

UCM Batch Manager Business Service

The user properties for this business service appear in the following table.

UCM Batch Manager User Property Name	Value
Account AutoMatch	CIFAccountUpsert;CIFAccountUpsertRs;IXMLOperation_UPSERT **Note:** IXMLOperation_UPDATE is also supported.
Account NoMatch	CIFAccountInsert;CIFAccountInsertRs;IXMLOperation_ADD
Contact AutoMatch	CIFContactUpsert;CIFContactUpsertRs;IXMLOperation_UPSERT
Contact NoMatch	CIFContactInsert;CIFContactInsertRs;IXMLOperation_ADD

UCM Batch Manager User Property Name	Value
FINCORP Account NoMatch	CIFFINCORPAccountInsert;CIFFINCORPAccountInsertRs;IXMLOperation_ADD
Household AutoMatch	CIFHouseholdUpsert;CIFHouseholdUpsertRs;IXMLOperation_UPSERT
Household NoMatch	CIFHouseholdInsert;CIFHouseholdInsertRs;IXMLOperation_ADD

UCM Survivorship Engine

The user properties for this business service appear in the following table.

UCM Survivorship Engine User Property Name	Value
BRCurrencyFldConstruct_Contact	Income:Income Currency Code,Income Exchange Date
DefaultAttrGrpName_Account	Default Account Attribute Group
DefaultAttrGrpName_Contact	Default Contact Attribute Group
DefaultAttrGrpName_Household	Default Household Attribute Group
ExemptFields_Account	Party Uld;Id
ExemptFields_Contact	Party Uld;Person Uld;Id
ExemptFields_Household	Party Uld;Id
ExternalBSIOInterface_Account	UCMAccountAttributeGroupMetaData
ExternalBSIOInterface_Contact	UCMContactAttributeGroupMetaData
ExternalBSIOInterface_Household	UCMHouseholdAttributeGroupMetaData
KnowledgeBaseName	ucmsurvivorshiprules
NULLValueOverride_Account	TRUE
NULLValueOverride_Contact	TRUE
NULLValueOverride_Household	TRUE

UCM Survivorship Engine User Property Name	Value
EntityInternalProcess<ChildObjectName>	TRUE

UCM Account Source Data and History Service

The user properties for this business service appear in the following table.

UCM Account Source Data and History User Property 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 the following table.

UCM Contact Source Data and History User Property Name	Value
Application Services Interface	Y
Internal Integration Object	UCMContactSourceDataAndHistory

UCM FINCORP Account Source Data and History Service

The user properties for this business service appear in the following table.

UCM FINCORP Account Source Data and History User Property Name	Value
Application Services Interface	Y
Internal Integration Object	UCMFINCORPAccountSourceDataAndHistory

UCM Household Source Data and History Service

The user properties for this business service appear in the following table.

UCM Household Source Data and History User Property Name	Value
Application Services Interface	Y
Internal Integration Object	UCMHouseholdSourceDataAndHistory

Configuring UCM Server Components for Data Management and Survivorship

Configure the UCM server components UCM Batch Manager and UCM Batch Publish Subscribe for data management and survivorship functionality by setting the parameter values for these components. Parameter values can be set in the Server Manager GUI or at the command-line interface. For more information on setting the server component parameters, see *Siebel System Administration Guide* . For information on these UCM server components, see *About Oracle Customer Hub (UCM) Server Components*. Use the following task to configure UCM server components for data management and survivorship.

To configure UCM server components for data management and survivorship

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

Parameter	Alias	Default Value	Description
UCM Batch Object Type	UCMBatchObjectType	Contact	Object Type for UCM Batch Publish Subscribe or UCM Batch Manager
UCM Batch Size	UCMBatchSize	10	Page Size for UCM Batch Publish Subscribe or UCM Batch Manager
UCM Data Management Cleanse Flag	UCMCDMCleanseFlag	False	Enables UCM Data Management Cleanse Capability
UCM Data Management Exact Match Flag	UCMCDMExactMatchFlag	False	Enables the UCM Data Management Exact Match Capability
UCM Data Management Match Flag	UCMCDMMatchFlag	False	Enables the UCM Data Management Match Capability
UCM Publish/Subscribe Flag	UCMPubSubFlag	False	Enables the UCM Publish/ Subscribe Capability

Parameter	Alias	Default Value	Description
UCM Search Specification	UCMSearchSpec	Not applicable	Search Specification
UCM Sleep Time	UCMSleepTime	60	Sleep Time (seconds)
UCM Sort Specification	UCMSortSpec	Not applicable	Sort specification
UCM Survivorship Engine Flag	UCMSurvivorshipEngineFlag	False	Enables the UCM Survivorship Engine Capability

Configuring Run-Time Events for UI Interaction

This topic describes how to configure run-time events for UI interaction. For background information on run-time events and on how to create them, see *Siebel Personalization Administration Guide* . Use the following task to configure run-time events for UI interaction.

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, which 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 view, and create four events with the following field information.

Object Type	Object Name	Event	Subevent	Action Set Name	Sequence
Applet	Contact List Applet	InvokeMethod	WriteRecord	UCM WriteRecord	1
Applet	Contact List Applet	PreInvokeMethod	WriteRecord	UCM PreWriteRecord	1
Applet	Contact List Applet	InvokeMethod	DeleteRecord	UCM DeleteRecord	1
Applet	Contact List Applet	PreInvokeMethod	DeleteRecord	UCM PreDeleteRecord	1

4. Create the same four events described in Step 3 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 so that Oracle Customer Hub Survivorship Engine can evaluate the data. Use the following task to configure a default system for UI interaction.

To configure a default system for UI interaction

- Register the system in the System Registration view with the following data:
  - System ID is set to Default System.
  - System Name is Default System.

  For more information on this task, see *Registering Systems Connected to Oracle Customer Hub (UCM)*.

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, which is one of the following:

- **Siebel EAI MQSeries Transport.** It enables you to integrate data between Siebel Business Applications and external applications that can communicate with the IBM MQSeries.
- **Siebel EAI JMS Transport.** It transports messages to and from IBM MQSeries queues. To configure Siebel EAI MQSeries Transport, see *Process of Configuring Inbound Communication Using Siebel EAI MQSeries Transport*.
- **Siebel EAI HTTP Transport.** It enables 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 *Process of Configuring Siebel EAI HTTP Transport*.

  For more information on the Siebel Connector for Master Data Applications, see *Oracle Customer Hub (UCM) Prebuilt Business Services*. For background information and details on configuring Siebel EAI MQSeries Transport and Siebel EAI HTTP Transport, see *Transports and Interfaces: Siebel Enterprise Application Integration* .

Process of Configuring Inbound Communication Using Siebel EAI MQSeries Transport

To configure Oracle Customer Hub (UCM) inbound communication using the Siebel EAI MQSeries Transport, perform the following tasks:

- *Configuring Named Subsystems*
- *Configuring MQSeries Server Receiver of Siebel Server Component*.

Note: Make sure Siebel EAI MQSeries Transport is enabled prior to configuring inbound communication. For more 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* .

Configuring Named Subsystems

This task is a step in *Process of Configuring Inbound Communication Using Siebel EAI MQSeries Transport*. Use the following task to configure the named subsystems.

To configure named subsystems

1. Navigate to the Administration-Server Configuration screen.
2. From the link bar, click Enterprises.
3. Click the Profile Configuration View tab.
4. Create two new records in the Profile Configuration list and provide the following information.

Field	Record 1	Record 2
Name	Use any name, for example, CIFMQConnSubsy.	Use any name, for example, CIFMQDataSubsys.
Alias	This field is required for creating this record.	This field is required for creating this record.
Subsystem Type	MQSeriesServerSubsys	EAITransportDataHandlingSubsys

Note: The subsystem type that you select must 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	The queue name where the inbound request message will be received.	Not applicable
MQSeries Queue Manager Name	The Queue manager name.	Not applicable
MQSeries Response Physical Queue Name	The queue name where response messages will be sent.	Not applicable
MQSeries Sleep Time	Set to 100 (or longer if required).	Not applicable

Parameter Name	Record 1	Record 2
Workflow Process to Execute	Not applicable.	UCM (Organization, Person, Financial Asset, or Group) Customer Profile Integration SOAP Process

6. Save both records.

   For information on this procedure and the named subsystems, see *Siebel System Administration Guide* .

Configuring MQSeries Server Receiver of Siebel Server Component

This task is a step in *Process of Configuring Inbound Communication Using Siebel EAI MQSeries Transport*. Use this task to configure an MQSeries Server Receiver of the Siebel Server component.

To configure an MQSeries Server Receiver of the 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. Click the Parameters View tab under the Components list.
7. Set the parameters listed in the following table.

Parameter Name	Alias	Value
Receiver Connection Subsystem	ReceiverConnectionSubsystem	The subsystem name (for example, CIFMQConnSubsys or CIFMQDataSubsys) that you created already. For more information, see *Configuring Named Subsystems*.
Receiver Method Name	ReceiverMethodName	ReceiveDispatch or ReceiveDispatchSend.
Default Tasks	DfltTasks	The value must be at least one (1) or the number of tasks that you want performed.

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

Process of Configuring Siebel EAI JMS Transport to Integrate Data

Use the Siebel EAI JMS Transport to integrate data between Siebel Business Applications and external applications that can communicate using the Java Message Service (JMS) Protocol. The EAI JMS Transport transfers messages to and from the JMS queues.

To configure inbound communication using the Siebel EAI JMS Transport, you must perform the following tasks:

- *Configuring Named Subsystems*
- *Configuring JMS Receiver Siebel Server Component*

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

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

Configuring Named Subsystems

This task is a step in *Process of Configuring Siebel EAI JMS Transport to Integrate Data*. Use this task to configure the named subsystems.

To configure named subsystems

1. Navigate to the Administration-Server Configuration screen.
2. From the link bar, click Enterprises.
3. Click the Profile Configuration View tab.
4. Create two new records in the Profile Configuration list and provide the following information.

Field	Record 1	Record 2
Name	Any name, for example, CIFJMSConnSubsy.	CIFJMSDataSubsys.
Alias	This field is required for creating this record.	This field is required for creating this record.
Subsystem Type	JMSSubsys.	EAITransportDataHandlingSubsys.

Note: The subsystem type that you select must 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
ConnectionFactory name	Name of Connection Factory	Not applicable
ConnectionUsername	User Name for the connection	Not applicable
ConnectionPassword	User Password for the connection	Not applicable
ReceiveQueue name	Queue name to receive inbound request message from	Not applicable
SendQueue name	Queue name to send response message to	Not applicable
JVM Subsystem name	Java	Not applicable
DispatchWorkflowProcess	Not applicable	UCM (Organization, Person, Financial Asset, or Group) Customer Profile Integration SOAP Process

> **Note:** Make sure that you have created a named subsystem called JAVA for the subsystem called JVMSubSys with defined parameters for the DLL, Classpath, and VMOPTIONS parameters.

For information on how to create this subsystem, see *Business Processes and Rules: Siebel Enterprise Application Integration* .

6. Save both records.

For information on this procedure and the named subsystems, see *Siebel System Administration Guide* .

Configuring JMS Receiver Siebel Server Component

This task is a step in *Process of Configuring Siebel EAI JMS Transport to Integrate Data*. Use this task to configure the JMS receiver Siebel Server component.

To configure JMS 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, choose a Siebel Server.
4. Click the Components View tab.
5. In the Components list, choose JMS Receiver (alias: JMSReceiver).
6. Click the Parameters View tab under the Components list.

7. Set the parameters in the following table.

Parameter Name	Alias	Value
Receiver Connection Subsystem	ReceiverConnectionSubsystem	Subsystem name created previously, for example, CIFJMSConnSubsy
Receiver Method Name	ReceiverMethodName	ReceiveDispatch or ReceiveDispatchSend

8. Restart the Siebel Server and make sure the JMS Receiver server component is running.

Process of Configuring Siebel EAI HTTP Transport

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

- *Configuring the Siebel Web Engine to Run Inbound HTTP Transport*
- *Configuring Oracle Customer Hub (UCM) to Run the Extension Service*

Note: Make sure Siebel EAI HTTP Transport is enabled before configuring.

For more 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* .

Configuring the Siebel Web Engine to Run Inbound HTTP Transport

This task is a step in *Process of Configuring Siebel EAI HTTP Transport*. Use this task to configure the Siebel Web Engine to run the Inbound HTTP Transport.

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

1. Open eapps_sia.cfg file in the `\bin` subdirectory in the installation directory.
2. Find the section `[/ucm_%language%]`, for example `[/ucm_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* .

Configuring Oracle Customer Hub (UCM) to Run the Extension Service

This task is a step in *Process of Configuring Siebel EAI HTTP Transport*. Use this topic to configure Oracle Customer Hub (UCM) to run the extension service.

To configure Oracle Customer Hub (UCM) to run the extension service

1. Open ucm.cfg file in the `\bin` subdirectory in the installation directory.
2. Find the section `[/HTTP Services]`. Or, add this section if it is not found.
3. Add the services and corresponding named subsystems, for example:

```
 [/HTTP Services]
SiebelCIFContact = CIFInboundHTTPDispatch
```

   The name, SiebelCIFContact, in the query string matches the name, CIFInboundHTTPDispatch, which in turn looks up the named subsystem list and executes the dispatch as required.
4. Save and close the configuration file.

About the Integration of Oracle Customer Hub (UCM) with Siebel Business Applications

Oracle Customer Hub (UCM) is preconfigured to support integration with a combined instance of Siebel Business Applications and Oracle Customer Hub (UCM). When enabled, the integration allows Oracle Customer Hub (UCM) to apply history, cross-reference, and data quality processes to Siebel CRM contact and account data.

The integration uses Oracle Customer Hub (UCM) interface to deliver and receive data messages. It 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 integration of Oracle Customer Hub (UCM) with Siebel Business Applications, see *Extending the Integration of Oracle Customer Hub (UCM) with Siebel Business Applications*.

To enable the preconfigured integration of Oracle Customer Hub (UCM) with Siebel Business Applications, make sure the system preference, Enable UCM Processes, is set to True. For more information on UCM System Preferences, see *About System Preferences for Oracle Customer Hub (UCM)*.

Extending the Integration of Oracle Customer Hub (UCM) with Siebel Business Applications

Extending the preconfigured integration of Oracle Customer Hub (UCM) with Siebel Business Applications requires the following high-level configurations:

- Updating the appropriate Integration Object
- Generating a new XSD
- Updating the Transformation XSLT
- Updating the Validation XSLT if required

For more information on these tasks, see *XML Reference: Siebel Enterprise Application Integration* .

Configuring Siebel Master Data Integration Objects

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

- *About Siebel Master Data Application Integration Objects*
- *Modifying Siebel Master Data Application Integration Objects*
- *Extending UCM Integration Objects*
- *Adding the Status Key Property to Oracle Customer Hub (UCM) Integration Objects*

About Siebel Master Data Application Integration Objects

There are three types of Oracle Master Data Management 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 application. Many preconfigured integration objects are provided, 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, using Integration Object Builder in Siebel Tools. For more information about this task, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

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 to a format understood by 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 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.

UCM Dispatcher Map Integration Objects

The UCM dispatcher map integration object 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.** Indicates the default dispatcher map used
- **CIFDispMapAccount.** Indicates the dispatcher map containing only CIFAccount messages
- **CIFDispMapContact.** Indicates the dispatcher map containing only CIFContact messages
- **CIFDispMapFINCORPAccount.** Indicates the dispatcher map containing only CIFFINCORPAccount messages
- **CIFDispMapHHMisc.** Indicates the dispatcher map containing the CIFHousehold, CIFFINCORP Account, and CIFPersonal Address messages
- **CIFDispMapPerf.** Indicates the dispatcher map containing a subset of CIFContact messages for improving system performance

For performance reasons in a production environment, deactivate the 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*.

To modify Siebel Master Data Application integration objects

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

Note: Modify the UCM integration objects by activating or deactivating only the integration component fields and integration components. A large customization is difficult to upgrade.

Extending UCM Integration Objects

This topic describes the task of extending integration objects 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 that is 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*.

To modify UCM integration objects to include other data

1. From Siebel Tools, choose Object Explorer, then 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* .

Adding the Status Key Property to Oracle Customer Hub (UCM) Integration Objects

This topic describes the task of adding the Status Key property to Oracle Customer Hub (UCM) integration objects. For background information on Siebel Master Data Application integration objects, see *About Siebel Master Data Application Integration Objects*.

To add the status key property to Oracle Customer Hub (UCM) integration object

1. Make sure that there is no active status object user property in the UCM Transaction Manager business service.

   For background information on UCM Transaction Manager, see *UCM Transaction Manager*.
2. Modify the input integration object to reflect the status key that you require at each integration component level.

   For more information on this task, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .
3. Set the Siebel Adapter user property, StatusObject, to True, using Siebel Tools, and deploy to Siebel Runtime Repository.

About the Exact Match Functionality

The exact match functionality is one of the features that you can enable to help prevent the creation of duplicate records. Oracle Customer Hub (UCM) applies the logic for determining whether an incoming record matches an existing record in the following sequence:

1. If the cross-reference match functionality is enabled, then Oracle Customer Hub (UCM) first searches for the ID of the record in the cross-reference table.

2. If the cross-reference search does not find a matching record, and if the exact match functionality is enabled, then Oracle Customer Hub (UCM) invokes the exact match functionality (Query By Example or Query By User Key).
3. If a matching record is not found by the cross-reference search or the exact match search, and if DQ Match is enabled, then Oracle Customer Hub (UCM) invokes DQ Match, using a third-party matching engine.

If none of these searches finds a matching record, then Oracle Customer Hub (UCM) inserts a new record with the data from the incoming record.

An exact match search is successful if a single matching record is returned. For the exact match functionality to return a single matching record, all of the following criteria must be met:

- All of the fields specified in the Name ExactMatch Field user property or properties for an object (the exact match fields) are present in the incoming record.

  > **Note:** If more than one exact match field is specified for an object, then the exact match functionality combines the exact match fields using the AND operator, and performs a single query for a matching record.

- All of the exact match fields have a value in the incoming record.
- The values in the exact match fields in the incoming record are the same as the values in the exact match fields in a single record in the database.

The incoming record is then handled as follows, depending on the result of the exact match search:

- **If the exact match search finds no matching record.** In this case, if DQ Match is not enabled, then a new record is inserted with the data from the incoming record. If DQ Match is enabled, then Oracle Customer Hub (UCM) invokes DQ Match.
- **If the exact match search returns a single matching record.** In this case, the incoming record is merged with the existing matching record.
- **If the exact match search finds multiple matching records.** In this case, if DQ match is not enabled, then the incoming record is flagged for manual review. If DQ Match is enabled, then Oracle Customer Hub (UCM) invokes DQ Match.

To reduce the risk of duplicate records being created, it is recommended that you do the following as a best practice:

- Deduplicate the data in the database to make sure that there are no duplicate records with the same values in all of the exact match fields.
- Use the exact match fields as user key columns at the database-table level.

Configuring Exact Match Functionality in Oracle Customer Hub (UCM)

You configure Oracle Customer Hub (UCM) exact match functionality by configuring user properties in the UCM Data Quality Manager business service. For information on configuring business service user properties using Siebel Tools, see *Integration Platform Technologies: Siebel Enterprise Application Integration* . Use the following task to configure exact match functionality.

To configure exact match functionality in Oracle Customer Hub (UCM)

- 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 + ";"

where:

- *Name* must be unique. It might be an integration object name, root component name, or supported objects, for example, account, contact.
- *IntObjName* is the integration object name. Use it for querying by user key.
- *ICFieldName* is the integration component field name.
- *BCFieldName* is the business component field name. Use it for querying by example.

If a single field can uniquely identify a record, then it is recommended that you use only that field as the exact match field. However, you can specify multiple exact match fields, separated by a semicolon (;). The exact match functionality combines the exact match fields using the AND operator, and performs a single query for a matching record.

An object can appear in only one ExactMatch Object property. You can create one ExactMatch Object property that includes all supported objects, or you can create one ExactMatch Object property for each object. Append a number to each ExactMatch Object property, even if you create only one.

The following table shows an example of configuring the exact match functionality for the UCM Data Quality Manager business service. In this example, the first ExactMatch Object user property is for the contact object, and the exact match fields are First Name, Last Name, Email Address, and Home Phone #. The exact match functionality returns a matching record if the values in these fields in a single record in the database match the values in the same fields in an incoming record. The second ExactMatch Object user property is for the account object, and the exact match fields are Name and Location.

User Property	Value
ExactMatch Object 1	Contact:CIFContactInsert;
Contact ExactMatch Field 1	First Name:First Name;Last Name:Last Name;Email Address:Email Address;Home Phone #:Home Phone #;
ExactMatch Object 2	Account:CIFAccountInsert;
Account ExactMatch Field 1	Name:Name;Location:Location;

You can create multiple Name ExactMatch Field properties for an object, and you can optionally append a number at the end of the user property names to distinguish them, for example, Contact ExactMatch Field 1, Contact ExactMatch Field

2, and so on. Each Name ExactMatch Field user property can contain a maximum of 250 characters. The exact match functionality concatenates all of the fields specified in all of the Name ExactMatch Field user properties for the object using the AND operator, and performs a single query for a matching record.

As an example, you might create two Name ExactMatch Field user properties for the contact object, as shown in the following table.

User Property	Value
ExactMatch Object 1	Contact:CIFContactInsert;
Contact ExactMatch Field 1	Social Security Number:Social Security Number;
Contact ExactMatch Field 2	Birth Date:Birth Date;

The exact match functionality concatenates the fields specified in these two user properties, and returns a matching record if the values in the Social Security Number and Birth Date fields in a single record in the database match the values in the same fields in an incoming record.

Note: You cannot use Name ExactMatch Field user properties to specify alternative queries for an object. For example, you cannot configure the exact match functionality to search first for a matching record using the fields specified in Contact ExactMatch Field 1, and if no matching record is found, then perform a second search for a matching record using the fields specified in Contact ExactMatch Field 2.

Configuring Siebel Data Quality Cleansing for Oracle Customer Hub (UCM)

Oracle Customer Hub (UCM) supports data cleansing, using the Siebel Data Quality module. To configure data cleansing for Oracle Customer Hub (UCM), see see 1515881.1 (Doc ID) *Siebel Data Quality Administration Guide* on My Oracle Support.

Configuring Data Matching for Oracle Customer Hub (UCM)

Oracle Customer Hub (UCM) supports data matching using Oracle Data Quality Matching Server. To configure this functionality for Oracle Customer Hub (UCM), see *Siebel Data Quality Administration Guide* .

Flags for Controlling Cleansing and Survivorship in Workflows

The following information lists the flags that enable you to control cleansing, matching and survivorship functionality in the UCM Person Customer Profile Integration SOAP Process and the UCM Organization Customer Profile Integration SOAP Process workflows.

Flag	Description
RealTimePubSub	Enables or disables publish and subscribe at the workflow level.
Turn on CDM Cleanse	Enables or disables Cleansing at the workflow level.
Turn on CDM Exact Match	Enables or disables Exact Match at the workflow level.
Turn on CDM Match	Enables or disables Matching at the workflow level.
Turn on CDM Survivorship	Enables or disables Survivorship at the workflow level.

Cleansing Batch Data and Matching Data

Batch processing provides a means to cleanse and match a large number of records from your base tables at one time. You can run batch jobs as stand-alone tasks or schedule batch tasks to run on a recurring basis. After the Data Quality Manager server component (DQMgr) is enabled and you have restarted the Siebel Server, you can start your data quality tasks.

You can start and monitor tasks for the Data Quality Manager server component by:

- Using the Siebel Server Manager command-line interface; that is, the srvrmgr program.
- Running Data Quality Manager component jobs from the Administration - Server Management screen, Jobs view in the application.

You can specify a data quality rule in the batch job parameters, which is a convenient way of consolidating and reusing batch job parameters and also of overriding vendor parameters. For more information about using the Siebel Server Manager and administering component jobs, see *Siebel System Administration Guide* . In particular, see the chapters about the Siebel Enterprise Server architecture, using the Siebel Server Manager user interface, and using the Siebel Server Manager command-line interface.

CAUTION: If you have written custom Siebel scripting for data matching on business components, such as Account, Contact, List Mgmt Prospective Contact, then the modifications of the fields that the script executes occur in the background, and they might not activate user interface features. A potential problem might be a script not triggering a UI feature, such as a window that shows a potential matching record.

Parameters in Data Quality Batch Jobs

The following table shows the parameters used in data quality batch jobs. The names of the parameters for both Data Quality Manager component jobs and srvrmgr commands are given.

Job Parameter or Server Manager Parameter	Required	Description
Business component name (Bcname)	Yes	The name of the business component. The values include:

Job Parameter or Server Manager Parameter	Required	Description
		• Account • Contact • List Mgmt Prospective Contact • Business Address. Applicable only to data cleansing operations. • For Siebel Industry Applications, the CUT Address designation is used instead of Business Address.
Business object name (bobjname)	Yes	The name of the business object. The values include: • Account • Contact • List Mgmt • Business Address. Applicable only to data cleansing operations.
Operation type (opType)	Yes	The type of operation. The values include: • **Data Cleansing.** Cleanses data. • **Key Generate.** Generates match keys. • **Key Refresh.** Updates match keys. • **DeDuplication.** Performs data matching.
Object Sorting Clause (objsortclause)	No	Applicable to data-matching operations only. Indicates how candidate records are sorted for optimal processing by the data-matching software. The default value is Dedup Token.

Object Where Clause (Objwhereclause)	No	Limits the number of records processed by a data quality task. Typically, you use the account's name or the contact's first name to split large, data quality batch tasks, using the first letter of the name. For example, the following object WHERE clause selects only French account records where the account name begins with *A*:
		`[Name] like 'A*' AND [Country] = 'France'`
		For example, the following object WHERE clause selects all records where Name begins with *Paris* or ends with *London:*
		`[Name] like 'Paris*' or [Name] like '*london'`

Data Quality Setting (DQSetting)	No	Specifies data quality settings for data cleansing and data matching jobs. This parameter has three values separated by commas: • **First value.** Applicable to Search Software America (SSA) only. Search Software America (SSA) is a data quality vendor which may be used. If this value is set to Delete, the existing duplicate records are deleted. Otherwise, the existing duplicate records are not deleted. This is the only usage for this value. • **Second value.** Applicable to the Universal Connector only. It specifies whether the job is a full or incremental data matching job.

Job Parameter or Server Manager Parameter	Required	Description
		• **Third value.** Applicable to Firstlogic data quality software only. Firstlogic is a data quality vendor which may be used. It specifies the name of a dataflow file. For more information about the use of the data quality setting, see *Data Quality Guide for Oracle Customer Hub (UCM)*.
Key Type	No	Specifies a value for the key type data quality parameter. It is applicable to SSA only. For more information about this parameter, see *Data Quality Guide for Oracle Customer Hub (UCM)*.
Search Type	No	Specifies a value for the search type data quality parameter. It is applicable to SSA only. For more information about this parameter, see *Data Quality Guide for Oracle Customer Hub (UCM)*.
Threshold	No	Specifies a value for the Threshold data quality parameter. It is applicable to SSA only. For more information, see *Data Quality Guide for Oracle Customer Hub (UCM)*.
Rule Name	No	Specifies the name of a data quality rule. A rule with the specified name must have been created in the Administration - Data Quality screen, for example: `RuleName="Rule_Batch_Account_Dedup"` For more information, see *Data Quality Administration Guide*.

Cleansing Data Using Batch Jobs

The following procedure describes how to use a batch job to perform data cleansing on records in a selected business component. To effectively exclude selected records when running data cleansing tasks, you must add the following command to your object WHERE clause:

```
[Disable DataCleansing] <> 'Y'
```

Note: When you run a process in batch mode, any visibility limitation against your targeted data set is ignored. It is recommended that you allow a limited number of resources to access the Siebel Server Manager to run your data quality tasks, otherwise you run the risk of corrupting data.

Use the following task to perform data cleansing using batch jobs.

To perform data cleansing using batch jobs

1. Start the Server Manager program.
2. At the srvrmgr prompt, enter of the commands in the following table to perform data cleansing tasks.

Business Component	Example of Server Manager Command
Acccount	`run task for comp DQMgr with bcname=Account, bobjname = Account, opType="Data Cleansing", objwhereclause="[field_name] LIKE 'search_string*'", DqSetting="'','','account_datacleanse.xml'"`
Business Address	`run task for comp DQMgr with bcname = "Business Address", bobjname="Business Address", opType="Data Cleansing", objwhereclause = "[field_name] LIKE 'search_string*'", DqSetting="'','', 'business_address_datacleanse.xml'"`
Contact	`run task for comp DQMgr with bcname=Contact, bobjname=Contact, opType="Data Cleansing", objwhereclause LIKE "[field_name]='search_string*'", DqSetting="'','','contact_datacleanse.xml'"`
List Mgmt Prospective Contact	`run task for comp DQMgr with bcname= "List Mgmt Prospective Contact", bobjname="List Mgmt", opType="Data Cleansing", objwhereclause LIKE "[field_name]='search_string*'", DqSetting="'','','prospect_datacleanse.xml'"`

15 Configuring Oracle Higher Education Constituent Hub

Configuring Oracle Higher Education Constituent Hub

This chapter contains an overview of Oracle Higher Education Constituent Hub. It contains the following topics:

- *About Oracle Higher Education Constituent Hub*
- *Viewing Constituent Data in Oracle Higher Education Constituent Hub*
- *Scenarios for Using Oracle Higher Education Constituent Hub*
- *Configuring Oracle Higher Education Constituent Hub*
- *Administering Oracle Higher Education Constituent Hub*

About Oracle Higher Education Constituent Hub

Oracle Higher Education Constituent Hub provides for the needs of higher education institutions, which typically have a large number of systems where a wide variety of personal, biographical and demographic data is captured. Oracle Higher Education Constituent Hub provides a means to maintain accurate, single source control of all these disparate data across your enterprise. Oracle Higher Education Constituent Hub enables you to capture, standardize and correct constituent names and addresses; identify and merge duplicate records; enrich constituent profiles; enforce compliance and risk policies; and distribute a best version record to all subscribing systems. For a list of schema changes for this release of Oracle Higher Education Constituent Hub, see *Schema Changes for Oracle Customer Hub (UCM) Version 8.2*.

What Is a Constituent?

A *constituent* is record that is analogous to a contact in Oracle Customer Hub (UCM). For example, in Oracle Higher Education Constituent Hub, constituents are defined as applicants, students, alumni, faculty, donors, and staff.

Viewing Constituent Data in Oracle Higher Education Constituent Hub

The Constituents screen displays all constituent views grouped under one parent category. The primary view is the Higher Education Constituent screen. It displays all relevant constituent information within your system. Use the following task to view data in Oracle Higher Education Constituent Hub.

To view data in Oracle Higher Education Constituent Hub

1. Log in to Oracle Customer Hub (UCM).

2. Navigate to the Administration - Universal Customer Master screen, then the Constituents view.
3. Select any constituent record to view basic biographical data, such as personal address, city, and so on.
4. Click the view tabs to access more information, such as other addresses, affiliations, other phone numbers, and so on.

Scenarios for Using Oracle Higher Education Constituent Hub

This topic gives the following example of how Oracle Higher Education Constituent Hub might be used. You might use Oracle Higher Education Constituent Hub differently, depending on your business model.

- *Adding a New Constituent in Oracle Higher Education Constituent Hub*
- *Updating a Constituent Record in Oracle Higher Education Constituent Hub*
- *Searching a Constituent Record in Oracle Higher Education Constituent Hub*

Adding a New Constituent in Oracle Higher Education Constituent Hub

An administrator using a source application, such as Oracle's PeopleSoft Campus Solutions, adds a new constituent record. The constituent has different names, addresses, phone numbers, and email addresses. At the time the record is saved, a create message is published to Oracle Higher Education Constituent Hub. Oracle Customer Hub (UCM) accepts the create message and processes it through the matching and survivorship rules, resulting in either the creation of a new constituent or a link and update of an existing constituent. The response message is sent synchronously to the requester. In parallel, a create-and-update person message is published to the subscribing systems.

Updating a Constituent Record in Oracle Higher Education Constituent Hub

A campus solutions user updates a constituent record. The updated record is then sent to Oracle Higher Education Constituent Hub. It evaluates the update and applies the survivorship rules prior to updating the constituent's best version record. The updated record is then sent back to the requesting system. In parallel, the update person message is published to all other subscribing systems.

Searching a Constituent Record in Oracle Higher Education Constituent Hub

Prior to creating a new constituent record in campus solutions, an administrator searches Oracle Higher Education Constituent Hub to avoid creating a duplicate ID. The search results return information from the Name, Address, Phone, Email types, or Effective Start Date fields. The match request is sent to the Higher Education Constituent Hub with the search parameters. Oracle Higher Education Constituent Hub returns a list of candidates with the matching scores for the user to select. A query request is then sent to Oracle Higher Education Constituent Hub for selected candidates, and

the candidate details are returned. If the candidate already exists in the requesting system, the cross-ref ID (for Campus Solutions, the ID is EMPLID) is returned. If the candidate does not exist in the system, the user can choose to import the data from Oracle Higher Education Constituent Hub in to the requesting system (Campus Solutions) and continue working on the new constituent.

Configuring Oracle Higher Education Constituent Hub

This topic describes how to configure Oracle Higher Education Constituent Hub.

To activate the constituent address functionality in Oracle Higher Education Constituent Hub

1. In the Siebel Tools object explorer, choose Integration Object.
2. In the Integration Objects screen, query for UCMContactSourceDataAndHistory.
3. In the Object Explorer, click Integration Component.
4. In the Integration Components view, query for UCM HE Constituent Address.
5. Deselect the Inactive field, recompile the project, and redeploy the integration object.
   For information on deploying integration objects, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

Administering Oracle Higher Education Constituent Hub

The following topics detail administration tasks for Oracle Higher Education Constituent Hub:

- *Creating New Constituent Records*
- *Editing Constituent Records*
- *Adding Additional Constituent Addresses*
- *Adding Affiliations to Constituent Records*
- *Creating Email Addresses for Constituents*
- *Entering Constituent Identification Numbers*
- *Entering Constituent Names*
- *Creating Constituent Phone Numbers*
- *Viewing All Constituent Data*
- *About External ID Records*
- *Viewing a Constituent's Historical Record*

Creating New Constituent Records

Use this task to create constituent records directly in Oracle Higher Education Constituent Hub.

To create a new constituent record

1. Navigate to the Administration - Universal Customer Master screen, then the Constituents view.
2. In the Constituents screen, click New to create a new record, or query for the record and make the changes.
3. From the application level menu, choose Save Record.

Editing Constituent Records

Use this task to edit constituent records in the Oracle Higher Education Constituent Hub Constituent screen.

To edit a constituent record

1. Navigate to the Administration - Universal Customer Master screen, then the Constituents view.
2. In the Constituents screen, query for the record, then click the More Info tab.
3. Add any new data or make any changes in the More Info view.
4. Some fields are defined in the following table.

Field	Comments
Name	Click the Select button, and choose or create alternate names for the constituent.
ID #	Click the Select button, and choose a form of ID, such as Passport, Social Security Number, and so on. If you are creating a new record, indicate whether the ID form you are entering is primary, then choose an ID type from the list, and complete the remaining fields, then click OK.
Veteran Benefit	Select the check box if the constituent is eligible for veteran's benefits.
Class Year	Click the Select button, and enter a numeric year.
Degree	From the list, select the degree that the constituent has attained.

5. After entering information, choose Save Record from the application-level menu.

Adding Additional Constituent Addresses

You add additional constituent addresses in the Address View tab in the Constituent screen. An additional address might be home, business, permanent, dormitory, campus, billing, mailing, and so on. Use the following task to add additional constituent addresses.

To add additional constituent addresses

1. From the Constituents screen, choose the constituent record, then click the Address tab.

2. Click New, and enter the new address information.
3. Select the Primary check box to designate a primary address.
4. Enter the remaining information, then choose Save Record from the application-level menu.

Adding Affiliations to Constituent Records

Affiliations indicate a person's status at an institution. Affiliation types are: student, staff, faculty, or alumnae. You can create all required affiliation records for a constituent. Use the following task to create affiliations for constituents.

To create affiliations for constituents

1. From the Constituents screen, choose the constituent record, then click the Affiliation tab.
2. Click New and enter the new affiliation information.
3. Click the Primary check box to designate a primary affiliation.
4. Enter the remaining information, then choose Save Record from the application-level menu.

Creating Email Addresses for Constituents

You enter email addresses, either primary or secondary, to a constituent record in the Email view.

To create email records for constituents

1. From the Constituents screen, choose the constituent record, then click the Email tab.
2. Click New, and enter the required information, some of which is shown in the following table.

Field	Description
Primary	Select this field to designate the email address as primary for the constituent record.
E-mail	Enter the email address.
Use Type	Click the list to designate the email address type. The values include: Home, Business, Campus, and so on.
Effective Start Date and Effective End Date	Click the Select button to choose an effective start and end date for the email record.
Description	Enter a description in this field.

3. From the Application-level menu, choose Save Record.

   Any changes you make to a constituent's email address are tracked in the Source Data History table. You can view these constituent email address history records in the Email history view.

Entering Constituent Identification Numbers

Constituent identification types can be: driver's license numbers, birth certificate numbers, social security numbers, any state, country or campus ID, and so on. You can enter all constituent identification numbers and types in the Identification view from the Constituent screen. Use the following task to enter constituent identification information.

To enter constituent identification

1. From the Constituents screen, choose the constituent record, then click the Identification tab.
2. Click New, and enter the required information, some of which is shown in the following table.

Field	Description
Primary	Select this field to designate the identification as primary for the constituent record.
ID Type	Click the list to select an identification type. Values are: Passport, SSN (Social Security Number), and Birth Number.
ID #	Enter the identification number.
Country	Click the list, and select a country.
Effective Start Date, and Effective End Date	Click the Select button to choose an effective start and end date for the email record.
Description	Enter a description in this field.

3. From the Application-level menu, choose Save Record.

Entering Constituent Names

Constituent names can be primary or alternate names, such as Given, Maiden, Married, Legal, Degree, Diploma names, and so on. Use the following task to enter constituent name information.

To enter constituent name information

1. From the Constituents screen, choose the constituent record, then click the Name tab.

2. Click New, and enter the required information, some of which is shown in the following table.

Field	Description
Primary	Select this field to designate the Name record as primary for the constituent's record.
Preferred First Name	Enter the constituent's preferred first name, such as nickname or diminutive.
Name Type	Click the list to designate a type for the name record, such as Given, Preferred, Alias, and so on.

3. From the Application-level menu, choose Save Record.

Creating Constituent Phone Numbers

You can create phone numbers, either primary or secondary, for a constituent record in the Phone view. Use the following task to create phone records for constituents.

To create phone records for constituents

1. From the Constituents screen, choose the constituent record, then click the Phone tab.
2. Click New, and enter the required information, some of which is shown in the following table.

Field	Description
Primary	Select this field to designate the phone number as primary for the constituent record.
Phone #	Enter the phone number.
Use Type	Click the list to designate the email address type. The values include: Home, Business, Campus, and so on.
Effective Start Date, and Effective End Date	Click the Select button to choose an effective start, and end date for the phone record.
Description	Enter a description in this field.

3. From the Application-level menu, choose Save Record.

Viewing All Constituent Data

You view all data collected from a constituent in the Summary view on the Constituent screen. This view enables you to view and edit all constituent data in one summary. Use the following task to view all constituent data.

To view all constituent data

1. From the Constituents screen, choose the constituent record, then click the Summary tab.

   The constituent record is displayed in the applets next to the main record.
2. Add any additional information about the constituent in the Address, Identification, Affiliation, Name, E-Mail, or Phone applet.

About External ID Records

Systems that are subscribed to Oracle Customer Hub (UCM) create records with their own record IDs. When messages are sent to Oracle Customer Hub (UCM), the external system ID is retained as well as the external record ID information, which is cross-referenced with the Oracle Customer Hub (UCM) record ID. The External ID View shows this cross-reference information of constituent records with different subscribing systems.

Viewing a Constituent's Historical Record

You can view the historical records of a constituent. Use the following task to view constituent historical data.

To view all constituent historical data

- From the Constituents screen, choose the constituent record, then click the Historical Version View tab.

  The constituent record is displayed with a list of all the historical records that has been added to the constituent record.

16 Oracle Automotive Captive Finance Customer Hub

Oracle Automotive Captive Finance Customer Hub

This chapter contains an overview of Oracle Automotive Captive Finance Hub. It contains the following topics:

- *About Oracle Automotive Captive Finance Customer Hub*
- *Oracle Automotive Captive Finance Customer Hub Views*

About Oracle Automotive Captive Finance Customer Hub

Oracle Automotive Captive Finance Customer Hub is a bundle of Oracle Customer Hub (UCM) products that provide a customer-hub solution for the automotive captive finance industry. It supports the mastering of automotive customers, including individual customers, businesses fleet owners and dealers along with various loan products offered, if any. It also stores vehicle details, policies, account transactions and special offers. Using Siebel Automotive, manufacturers, importers, distributors, and dealers can manage the needs of retail and fleet customers including customers financing, leasing, and insurance options. It captures all the information obtained from the factory-to-dealer-to-consumer value chain to provide a better understanding of customers and their needs.

Oracle Automotive Captive Finance Customer Hub Views

This topic lists the Oracle Automotive Captive Finance Customer Hub views that are additional to the base Oracle Customer Hub (UCM) views.

List of the additional Universal Customer Master - Base views in Oracle Automotive Captive Finance Customer Hub.

- SIS Account Administration View
- Relationship Hierarchy View (Account)
- Pharma Activity Chart View - Activity Analysis
- Pharma Activity Chart View - Status Analysis
- Relationship Hierarchy View (Contact)
- Households Across Organization View
- Relationship Hierarchy View (Household)

- Relationship Hierarchy View (Employee)
- CIF System Administration List View
- CIF System Privileges Admin List View
- CIF System Pub Sub Admin List View
- CIF Company References Admin List View
- CIF Contact References Admin List View
- CIF Household References Admin List View
- CIF Home Page View
- UCM Account History Flat List View
- UCM Account History List View
- UCM Account Cross-Reference View
- UCM Contact History Flat List View
- UCM Contact History List View
- UCM Contact Cross-Reference View
- UCM Contact DeDup View
- UCM Account DeDup View
- UCM Household Cross-Reference View
- UCM Attribute Group Field View
- UCM Rule Set Attribute Group Field View
- UCM Rule Set Attribute Group View
- UCM Rule Set Group Source Confidence View
- UCM Account Duplicates Detail View
- UCM Account Duplicates View
- UCM Account UnMerge View
- UCM Contact Duplicates Detail View
- UCM Contact Duplicates View
- UCM Contact UnMerge View
- UCM Account Contact List View
- UCM Account Privacy History List View
- UCM Contact Privacy History List View
- UCM Financial Account Admin List View
- UCM Financial Account Contact List View
- UCM Financial Account Privacy History List View
- UCM Financial Account Privacy List View
- UCM Household Contact View
- UCM Household History List View

- UCM Household History Flat List View
- UCM Hierarchy Flat List View
- UCM Hierarchy Tree Explorer View
- Contact Details View (Detail tab)
- Contact Detail - Personal Address View
- Relationship Hierarchy View (Contact)
- Contact Detail - Personal Payment Profile View
- Households Listing View
- Households Contact Detail View
- Relationship Hierarchy View (Household)
- All Channel Partner Across Organization
- Channel Partner Detail View
- Channel Partner Detail - Contacts View
- Channel Partner Address View
- Partner Partnership Contacts View
- Business Service List View
- Business Service Methods List View
- Business Service Script Editor View
- Business Service Test View
- Business Service User Properties View

List of the additional Universal Customer Master - FINS views in Oracle Automotive Captive Finance Customer Hub.

- FINS Company Financial Accounts View
- FINS Organization Group Policies View
- Account Policies Quotes View
- FIN Contact Account View
- INS Contact Detail - Policy View
- INS Household Detail - Policy View
- FINS Household Accounts View
- FINS AG All Agents View
- FINS AG All Agents across Organization
- FINS AG Agent More Info
- FINS AG Agent's Policy View

List of the additional Universal Customer Master - AUTO views in Oracle Automotive Captive Finance Customer Hub.

- Auto Contact Detail - Household View
- Auto Contact Image View
- Auto Contact Summary View

- Auto Contact Vehicle View
- Auto Account Vehicles View
- eAuto Households Auto Vehicle View (read-only)
- eAuto Households Summary View
- Auto Dealer Contact View
- eAuto Dealer Address View
- eAuto Dealer Detail View
- Auto Dealer Makes View

List of the additional Product Information File - Base views in Oracle Automotive Captive Finance Customer Hub.

- CIF System Administration List View
- CIF System Privileges Admin List View
- CIF System Pub Sub Admin List View
- CIF Company References Admin List View
- CIF Contact References Admin List View
- CIF Household References Admin List View
- CIF Home Page View
- UCM Account History Flat List View
- UCM Account History List View
- UCM Account Cross-Reference View
- UCM Contact History Flat List View
- UCM Contact History List View
- UCM Contact Cross-Reference View
- UCM Contact DeDup View
- UCM Account DeDup View
- UCM Household Cross-Reference View
- UCM Attribute Group Field View
- UCM Rule Set Attribute Group Field View
- UCM Rule Set Attribute Group View
- UCM Rule Set Group Source Confidence View
- UCM Account Duplicates Detail View
- UCM Account Duplicates View
- UCM Account UnMerge View
- UCM Contact Duplicates Detail View
- UCM Contact Duplicates View
- UCM Contact UnMerge View
- UCM Account Contact List View

- UCM Account Privacy History List View
- UCM Contact Privacy History List View
- UCM Financial Account Admin List View
- UCM Financial Account Contact List View
- UCM Financial Account Privacy History List View
- UCM Financial Account Cross-Reference View
- UCM Household Contact View
- UCM Household Contact View
- UCM Household History List View
- UCM Household History Flat List View
- UCM Hierarchy Flat List View
- UCM Hierarchy Tree Explorer View
- Category Master Data View - Products
- ISS Product Structure Designer View
- ISS Product Property Designer View
- Complex Product Runtime Instance View (Validation)
- ISS Product UI Designer View
- ISS Product UI Designer View2
- ISS Product Versions View
- FS Field Service Details Admin
- FS Product Measurements
- Internal Product Image View
- Mktg Admin Product Entitlement Templates View
- ISS Product Administration View
- ISS Joint Workspace View
- Product Detail Components Administration View
- Product Detail Key Features Administration View
- Product Detail Price Lists Administration View
- Product Detail Product Comparison Administration View
- Product Literature Administration View
- Product Translation Administration View
- Product XA Administration View
- FS Product Measurement (read-only) View
- FS Product Warranties View
- FS Product Measurement (read-only) View
- FS Product Warranties View

- Price Book Attributes Admin View
- Price Book Item Maker Single Admin View
- Price Book Items Admin View
- Price Book List View
- Product Attachment View
- Product Detail Components View
- Product Detail Cost Lists View
- Product Detail Key Features View
- Product Detail Price Lists View
- Product Detail Product Comparison View
- Product Detail Revenue Schedule Chart View
- Product Entitlement Templates View
- Product Inventory Location View
- Product List More Info View
- Product List View
- Product Literature View
- Product Organization View
- Product Related Defects List View
- Product XA View

List of the additional Product Information File - FINS views in Oracle Automotive Captive Finance Customer Hub.

- FINS Application Product Navigation View
- FINS Product Details SF View
- FINS Product Details SF View-Rates
- FINS Product News View
- FINS Product Scoring View
- FINS Product Details SF Agent View
- FINS Product Details SF Agent View-Rates
- FINS Product Details SF View
- FINS Product Details SF View-Rates
- FINS Product News Agent View
- FINS Product News View
- FINS Product Scoring View

List of the additional Universal Dealer Master - AUTO views in Oracle Automotive Captive Finance Customer Hub.

- eAuto Dealer Operation - Activities View
- eAuto Dealer Operation - Contact View
- eAuto Dealer Operation - Employee View

- eAuto Dealer Operation - Makes View
- eAuto Dealer Operations - Sales History View
- eAuto Dealer Operations - Service History View
- eAuto Dealer Operations Detail - Opportunity View
- eAuto Dealer Operations Detail - Order View
- eAuto Dealer Operations Detail - Service Request View
- eAuto Dealer Operations Detail - Quote View
- eAuto Dealer Operations Detail- Account View
- eAuto Dealer Operations List View
- eAuto Dealer Operations Sales Service Calendar View
- eAuto Dealer Organization Vehicle View
- eAuto Dealer Related Vehicle View
- eAuto Dealer Vehicle Inventory View
- eAuto All Dealer Operations List View
- eAuto Dealer Operations List View - More Info
- eAuto Manager's Dealer Operations List View
- eAuto My Dealer Operations List View
- Auto All Dealer List View
- Auto All Dealers Across Organization
- Auto Dealer Activities View
- Auto Dealer Activity Plan
- Auto Dealer Attachment View
- Auto Dealer Category View
- Auto Dealer Contact View
- Auto Dealer Makes View
- Auto Dealer Note View
- Auto Dealer Opportunities View
- Auto Dealer Profile View
- Auto Dealer Sales Service Calendar View
- Auto Manager Dealer List View
- Auto My Dealer List View
- eAuto Dealer Address View
- eAuto Dealer Agreement List View
- eAuto Dealer Assessment View
- eAuto Dealer Certification - Certifications Detail View
- eAuto Dealer Chart Over Metric View

- eAuto Dealer Chart Over Time View
- eAuto Dealer Detail View
- eAuto Dealer Fund Request View
- eAuto Dealer Organization Analysis View
- eAuto Dealer Revenue Chart View
- eAuto Dealer Revenue View
- eAuto Dealer Training Course View
- Dealers Screen Home Page View

List of the Universal Vehicle Master - AUTO views in Oracle Automotive Captive Finance Customer Hub.

- Auto All Vehicles across Organizations View
- Auto Vehicle Contacts View (SDW)
- Auto Vehicle Detail View
- Auto Vehicle Options View
- Auto Vehicle Summary View
- Auto Vehicle Tracking View
- eAuto Financial Detail View
- eAuto All Vehicle View
- eAuto My Vehicle View
- eAuto Vehicle Access Team View
- eAuto Vehicle Activities View
- eAuto Vehicle Attachment
- eAuto Vehicle Drivers View
- eAuto Vehicle Features View
- eAuto Vehicle Internal Division View
- eAuto Vehicle Manager's View
- eAuto Vehicle Measurements View
- eAuto Vehicle PM History View
- eAuto Vehicle PM Plan Items View
- eAuto Vehicle Readings View
- eAuto Vehicle Specifications View
- eAuto Vehicle Sub-Components View
- eAuto Vehicle Transaction View
- Auto Vehicle Admin View
- eAuto Vehicle Financial Accounts View

17 Oracle Case Hub

Oracle Case Hub

This chapter contains an overview of Oracle Case Hub and information on views that are additional to the default Oracle Customer Hub (UCM) views. It contains the following topics:

- *About Oracle Case Hub*
- *Oracle Case Hub Views*

About Oracle Case Hub

Oracle Case Hub is a bundle of Oracle Customer Hub (UCM) products which focus primarily on case management, such as, taxpayers, claimants, and so on.

Along with the mastering citizen, claimant, taxpayer data, and so on, Oracle Case Hub also supports the mastering of cases (insurance, justice and safety, revenue and tax, social services, transportation, and so on). It also supports the related activities with the optional details that depend upon the nature of cases like the following:

- Vehicle data, public-health related data, for example, disease, outbreaks, medication
- Justice and safety related data, for example, suspect, incident, arrest, evidence, leads, and so on

Main Features of Oracle Case Hub

The following are the main features of Oracle Case Hub.

Case Serialization

Case serialization provides agencies with the ability to standardize case numbering across the organization by generating unique, custom-defined serial numbers on cases, as well as leads and evidence that are related to the case. When a case record is created in the Siebel database, a unique serial number is automatically generated and stamped on the record. Using Siebel Case Serialization, administrators can set the case serial number to include a combination of multiple fields so that the composite number is meaningful to an investigator. For interoperability with non-Siebel data, case serial numbering can be accessed from other external applications using prebuilt Web services to extend an organization's standardized numbering methodology to case information in any system.

Incident Management

Siebel Incident Management allows agencies to track, monitor, escalate, and resolve reported incidents. Currently, most agencies manage reported incidents through either paper-intensive manual processes or stand-alone, nonintegrated systems. Procedures and processes for recording and routing incidents are inconsistent, time-consuming and error prone. Siebel Incident Management allows agencies to standardize the incident data capture process inside or across departments and ensure that incidents are resolved according to established department policies or regulatory requirements. Siebel Incident Management provides the flexibility to track an individual across multiple incidents, even when this individual has a different role in the incident. Siebel Incident Management enables organizations to track offenders, victims, or witnesses from incident to incident. This flexibility allows organizations to quickly gather the

information required to understand the context of an incident, route it to the appropriate department or individual for response, and ensure that the right course of action is pursued according to the specific policy and regulatory requirements of the agency.

Offense Management

Siebel Offense Management provides organizations the capability to track criminal offenses as part of a comprehensive incident management system. Offenses can be logged as part of the incident and used for reporting purposes and further analysis. Data about the offenders, victims, force used, and assets can be captured quickly and easily. Users can follow up on the offense by adding notes or related documents to the record to provide a comprehensive picture of the offense. All of the necessary information for reporting, tracking, and managing offenses are easily entered and readily available to users. Agencies can capture the circumstances surrounding the incident of specific categories, such as aggravated assault or murder, justifiable homicide, or negligent manslaughter with descriptive and structured information for ease in reporting and analyzing. Additionally, Siebel Offense Management allows agencies to track the use of drugs or alcohol related to the offense and record any use of force during the offense, including the use of any weapons.

Location Tracking

Siebel Location Tracking enables organizations to standardize locations that have multiple addresses and to specify an area where there might not be precise address information. The ability to assign more specific location information without needing addresses makes it easier to classify and capture important information about an area. Siebel Location Tracking supports the association of multiple addresses to a single, specific Location, as well as other precise location information, such as latitude-longitude pairs or Universal Transverse Mercator (UTM). This feature provides organizations with a variety of different address formats to use when describing the location of an incident (For example, Central Park, corner of 5th and Main Streets, parking lot G). This combination of a specific location with a descriptive label helps faster response to developing incidents in addition to building a richer repository of data for use in later analysis. Siebel Location Tracking also enables agencies to standardize locations that have multiple addresses, such as malls, hotels, office parks or campuses, and specify an area that lacks precise address information, such as an intersection, or the north end of a lake.

Evidence Management

Siebel Evidence Management enables agencies to track and manage evidence associated with the cases they are managing. Siebel Evidence Management supports a two-tier evidence model that enables agencies to track unique physical items in addition to managing these items as evidence in the context of one or more cases. For investigative cases, evidence collected at a crime scene (such as a weapon) can be viewed as a unique asset in Siebel CRM, and later converted to evidence in multiple cases. For each evidence record, different activities and documents can be tracked, in accordance with the specific requirements of each case. For social services or benefits cases, a person's identifying documents can serve as evidence in multiple cases across departments or programs. Siebel Evidence Management provides the capability to consolidate evidentiary items across an organization, yet still manage evidence in the context of each individual case. Siebel Evidence Management enables agencies to manage the history of activities or actions specific to each case and each piece of evidence. This feature includes the ability to establish standardized handling instructions or activity templates that conform to agency requirements. Agencies can also link multiple evidence items with the same underlying asset, or relate evidence items within or between cases to establish key investigative linkages. Additionally, Siebel Evidence Management enables agencies to track various roles of individuals as they relate to each piece of evidence. Together, these new capabilities in Siebel Evidence Management provide agencies with a flexible repository of physical items that supports evidence in different cases and at the same provides exceptional investigative insight.

Investigative Lead Management

This feature enables agencies to track lead information as part of each case record, including when the lead was created, routed, covered, and submitted for approval. Additionally, agencies can relate all relevant investigative

information to a lead, such as contacts, suspects, victims or other persons or groups of interest, and view and compile evidence associated with a lead that is relevant to the case at hand. For standardization, agencies can serialize leads as part of a case, and lead records can be numbered according to the specific numbering methodology of a public agency. When viewing the lead, agencies can also see activities performed on a lead automatically in the context of the parent case.

Oracle Case Hub Views

The following information lists the Oracle Case Hub views that are additional to the base Oracle Customer Hub (UCM) views.

Logical Grouping	Available Views
Public Sector Information File	HLS Account Case View
	HLS All Case List View
	HLS Case Account View
	HLS Case Activities View
	HLS Case Attachment View
	HLS Case Auto Vehicles View
	HLS Case Detail View
	HLS Case Disease View
	HLS Case Groups View
	HLS Case INS Claims View
	HLS Case Note View
	HLS Case Partner View
	HLS Case Preview View
	HLS Case Project View
	HLS Case Quote View

Logical Grouping	Available Views
	HLS Case Related Case View
	HLS Case Sales Assessment View
	HLS Case Service Request View
	HLS Case Suspect View
	PUB Case Audit Trail View
	PUB GOV Case Activity Plans View
Public Sector Information File (continued)	PUB GOV Case Monthly Calendar View
	PUB GOV Case Products View
	PUB GOV Channel Partner Cases View
	PUB GOV Contact Cases View
	PUB GOV Household Cases View
	PUB HLS Case Contact View
	PUB My Team Case List View
	PUB Case Screen Home Page View
	HLS Case Activities View - ReadOnly
	HLS Case Attachments View - ReadOnly
	HLS Case Detail View - ReadOnly
	HLS Case Hierarchy View
	HLS Case Note View - ReadOnly
	HLS Case Suspect View - ReadOnly
	HLS Case Teamspace View

Logical Grouping	Available Views
	PUB Service Request Case View
Public Sector Case Incident Information File	Disease Attachment View
	Disease Cases View
	Disease List View
	Disease Medication View
	Disease Service Request View
	Disease Training View
	Diseases Detail View
	HLS All Groups List View
	HLS All Medication List View
	HLS All Suspect List View
	HLS Disease FAQ View
	HLS Disease Preview View
	HLS Group Preview View
	HLS Groups Activities View
Public Sector Case Incident Information File (continued)	HLS Groups Address View
	HLS Groups Attachment View
	HLS Groups Case View
	HLS Groups Detail View
	HLS Groups List View
	HLS Groups Note View

Logical Grouping	Available Views
	HLS Groups Suspect View
	HLS Medication Diseases View
	HLS Medication Drug Interactions View
	HLS Medication Inventory View
	HLS Medication List View
	HLS Medication Substitutes View
	HLS Public Health Home Page View
	HLS Report Outbreak View
	HLS Smart Script Player View
	HLS Suspect Activities View
	HLS Suspect Address View
	HLS Suspect Associated Suspect View
	HLS Suspect Attachment View
	HLS Suspect Case View
	HLS Suspect Details View
	HLS Suspect Group View
	HLS Suspect List View
	HLS Suspect Note View
	HLS Suspect Preview View
	My Disease List View
	PUB HLS All Incident List View

Logical Grouping	Available Views
	PUB HLS Contact Incident View
	PUB HLS Groups Incident View
	PUB HLS Incident Activities View
	PUB HLS Incident Attachment View
Public Sector Case Incident Information File (continued)	PUB HLS Incident Detail View
	PUB HLS Incident Group View
	PUB HLS Incident List View
	PUB HLS Incident Note View
	PUB HLS Incident Preview View
	PUB HLS Incident Suspect View
	PUB HLS My Team Group List View
	PUB HLS My Team Incident List View
	PUB HLS My Team Suspect List View
	PUB HLS Suspect Incident View
	Relationship Hierarchy View (Groups)
	PUB Incident Screen Home Page View
	PUB HLS Case Incident View
	PUB HLS Exception Handling View
	PUB Incident Account Offense View
	PUB Incident Accounts View
	PUB Incident Arrest Activities View

Logical Grouping	Available Views
	PUB Incident Arrest Attachments View
	PUB Incident Arrest Force View
	PUB Incident Arrest Notes View
	PUB Incident Arrest Offense View
	PUB Incident Arrests View
	PUB Incident Audit Trail View
	PUB Incident Cases View
	PUB Incident Circumstances View
	PUB Incident Contacts View
	PUB Incident Evidence View
	PUB Incident Lead View
	PUB Incident Location View
	PUB Incident Offender Notes View
	PUB Incident Offender Offense View
Public Sector Case Incident Information File (continued)	PUB Incident Offender Victims View
	PUB Incident Offender View
	PUB Incident Offense Attachments View
	PUB Incident Offense Force View
	PUB Incident Offense Notes View
	PUB Incident Offense Offender View
	PUB Incident Offense Property View

Logical Grouping	Available Views
	PUB Incident Offense Substance View
	PUB Incident Offense Victim View
	PUB Incident Offenses View
	PUB Incident Property View
	PUB Incident Service Requests View
	PUB Incident Subject Matches View
	PUB Incident Subject View
	PUB Incident Victim Injuries View
	PUB Incident Victim Notes View
	PUB Incident Victim Offenders View
	PUB Incident Victim Offense View
	PUB Incident Victims View

18 Oracle Life Sciences Customer Hub

Oracle Life Sciences Customer Hub

This chapter contains an overview of Oracle Life Sciences Customer Hub and information on views that are additional to the base Oracle Customer Hub (UCM) view. It contains the following topics:

- *About Oracle Life Sciences Customer Hub*
- *Oracle Life Sciences Customer Hub Views*

About Oracle Life Sciences Customer Hub

Oracle Life Sciences Customer Hub is a bundle of Oracle Customer Hub (UCM) products that focus on the life sciences industry. It represents the most complete data model and baseline functionality to meet the customer relationship management needs of the pharmaceutical, biotechnology, medical devices, and clinical research industries. Oracle Life Sciences Customer Hub also helps pharmaceutical and medical product companies to maintain important data about managed care organizations and pharmacy benefit management companies.

This product provides a central repository for the following:

- **Contact management.** It enables users to access and manage all customer-related information for target contacts, including physicians, nurses, and other health-care professionals. Comprehensive profiling information, such as rating and ranking, multiple addresses, account and contact affiliations, activities, best times to visit, private and shared notes, and attachments, enable sales, marketing, and service contacts to maximize customer interaction. Users can share contact profile information with other sales team members, enabling productive team selling, and collaboration.
- **Complex account management.** It supports complex institutional sales process by mastering the complex account hierarchies that exist among hospitals, clinics, and various other organizations that influence purchasing decisions, such as Group Purchasing Organizations and Integrated Delivery Networks. Siebel Life Sciences enables the management of complex account organizational structures by supporting multilevel account hierarchies and intuitive drill-down capabilities. This feature enables account managers to conduct business and manage account and contact information at any level in the account hierarchy. Comprehensive account profiling information enable sales, marketing, and service contacts to maximize customer interaction. Users can share account profile information with other sales team members, enabling productive team selling and collaboration.
- **Account management for managed care organizations and pharmacy benefit management companies.** These businesses are entered as accounts of the type, Managed Care. The fields in the Managed Care Account form are designed to hold data relevant to these businesses, such as enrollment, prescription spending, and number of participating providers. Along with the mastering of physician, channel partner data, Oracle Life Sciences Customer Hub also stores pharmacy campaigns and offers.

Oracle Life Sciences Customer Hub Views

The following information lists the Oracle Life Sciences Customer Hub views that are additional to the base Oracle Customer Hub (UCM) view.

Logical Grouping	Available Views
Universal Customer Master - Base	SIS Account Administration View
	Relationship Hierarchy View (Account)
	Pharma Activity Chart View - Activity Analysis
	Pharma Activity Chart View - Status Analysis
	Relationship Hierarchy View (Contact)
	Households Across Organization View
	Relationship Hierarchy View (Household)
	Relationship Hierarchy View (Employee)
	CIF System Administration List View
	CIF System Privileges Admin List View
	CIF System Pub Sub Admin List View
	CIF Company References Admin List View
	CIF Contact References Admin List View
	CIF Household References Admin List View
	CIF Home Page View
	UCM Account History Flat List View
	UCM Account History List View

Logical Grouping	Available Views
	UCM Account Cross-Reference View
	UCM Contact History Flat List View
	UCM Contact History List View
	UCM Contact Cross-Reference View
	UCM Contact DeDup View
	UCM Account DeDup View
	UCM Household Cross-Reference View
	UCM Attribute Group Field View
	UCM Rule Set Attribute Group Field View
Universal Customer Master - Base (continued)	UCM Rule Set Attribute Group View
	UCM Rule Set Group Source Confidence View
	UCM Account Duplicates Detail View
	UCM Account Duplicates View
	UCM Account UnMerge View
	UCM Contact Duplicates Detail View
	UCM Contact Duplicates View
	UCM Contact UnMerge View
	UCM Account Contact List View
	UCM Account Privacy History List View
	UCM Contact Privacy History List View
	UCM Financial Account Admin List View

Logical Grouping	Available Views
	UCM Financial Account Contact List View
	UCM Financial Account Privacy History List View
	UCM Financial Account Privacy List View
	UCM Financial Account Cross-Reference View
	UCM Household Contact View
	UCM Household History List View
	UCM Household History Flat List View
	UCM Hierarchy Flat List View
	UCM Hierarchy Tree Explorer View
	Contact Details View (Detail tab)
	Contact Detail - Personal Address View
	Relationship Hierarchy View (Contact)
	Contact Detail - Personal Payment Profile View
	Households Listing View
	Households Contact Detail View
	Relationship Hierarchy View (Household)
	All Channel Partner Across Organization
	Channel Partner Detail View
	Channel Partner Detail - Contacts View
Universal Customer Master - Base	Channel Partner Address View
	Partner Partnership Contacts View

Logical Grouping	Available Views
	Business Service List View
	Business Service Methods List View
	Business Service Script Editor View
	Business Service Test View
	Business Service User Properties View
Advanced Customer Profile – Base	Account Detail - Activities View
	Account Agreement List View
	Account Entitlement List View
	Account Note View
	Sub Account Detail - Assets View
	Sub Account Detail - Contacts View
	Sub Account Detail - Sub Accounts View
	Sub Account Detail View
	Coverage Team View
	Contact Detail View
	Contact Service Agreement List View
	Contact Implicit Entitlements View
	Contact Explicit Entitlements View
	Contact Note View
	Households Notes View (read-only)
	Households Activities View (read-only)

Logical Grouping	Available Views
	FINS Household Detail - Billing Accounts View
	Households Service Agreements View (read-only)
	Channel Partner Detail - Activities View
	Channel Partner Agreement List View
	Partner Certification - Certifications Detail View
	Channel Partner Note View
	Channel Partner Profile View
Advanced Customer Profile – Base (continued)	Channel Partner Partnership Profile View
Marketing Information File – Base	Campaign Monthly Calendar View
	Campaign Achievement Chart View
	Campaign Activity View
	Campaign Analysis - Call Status
	Campaign Contacts View
	Campaign Detail
	Campaign Detail List
	Campaign Explorer View
	Campaign Lead
	Campaign Literature View
	Campaign Overview
	Contact Campaign View - Non Admin
	All Offers View

Logical Grouping	Available Views
	Direct Mail Offer Detail View
	Direct Mail Offer Literature View
	Direct Mail Offer View
	Fax Offer Detail View
	Fax Offer Literature View
	Fax Offer View
	Fax Profile Parameter View
	Indirect Offer Literature View
	Indirect Offer View
	Offer Files List
	Phone Offer Detail View
	Phone Offer Literature View
	Campaign Responses View - Non Admin
	Contact Response View - Non Admin
	Response Attachment View
Marketing Information File – Base (continued)	Response Detail View (Detail)
	Response Detail View
	Response Opportunity View
	Response Order View
	Response Product_Int View
	Response View

Logical Grouping	Available Views
	Campaign Responses View
Sales Information File – Base	Contact Detail - Opportunities View
	Households Opportunities Listing View (read-only)
	All Opportunity Attachments Across Orgs View
	All Opportunity Attachments View
	Contact Detail - Opportunities View
	My Clients Opportunities View
	My Company Opportunities View
	My Opportunity Attachments View
	My Teams Opportunity Attachments View
	Opportunity Default Chart View
	Opportunity Detail - Decision Issues View
	Opportunity Detail - Competitors View
	Opportunity Category Search View
	Opportunity Product Analysis Chart View
	All Opportunities across My Organizations
	All Opportunities across Organizations
	All Opportunity List View
	Manager's Opportunity List View
	Opportunity Administration View
	Opportunity Attachment View

Logical Grouping	Available Views
	Opportunity Category View
	Opportunity Detail - Activities View
	Opportunity Detail - Contacts View
Sales Information File – Base (continued)	Opportunity Detail - Indirect Accounts View
	Opportunity Detail - Organizations View
	Opportunity Detail - Products View
	Opportunity Detail - Sales Team View
	Opportunity Details
	Opportunity Explorer View
	Opportunity List View
	Opportunity Manager's Explorer View
	Opportunity Messaging List View
	Opportunity Note View
	Opportunity Organization Analysis View
	Opportunity Activity Plan
	Opportunity Assessment View
	Opportunity Detail - Competitors View
	Opportunity Detail - Decision Issues View
	Opportunity Skill View
	Opportunity Monthly Calendar View
	Opportunity Default Chart View

Logical Grouping	Available Views
	Account Quote View
	Admin Cost List
	Admin Cost List Item
	All Products across Organizations
	All Quote List View
	All Quotes across Organizations
	Contact Detail - Quote View
	Manager's Quote List View
	Opportunity Detail - Products View
	Opportunity Quote View
	Payment Terms
	Personal Quote List View
	ISS Product Administration View
Sales Information File – Base (continued)	ISS Joint Workspace View
	Product Detail Price Lists Administration View
	Product Line Administration View
	Quote Administration View
	Quote Attachment View
	Quote Bill To/Ship To View
	Quote Detail View
	Quote Item Detail View

Logical Grouping	Available Views
	Quote Item XA View
	Quote Item Promotion Deal View
	Quote List Detail View
	Quote List View
	Quote Manager's Explorer View
	Quote Orders View
	Quote Single Product View
	Quote Terms View

19 Embedded EDQ Access

Embedded EDQ Access

This chapter covers embedded EDQ Access. It contains the following topics:

- *About Embedded EDQ Access*
- *Setting Up Embedded EDQ Access*
- *Using Embedded EDQ Access*

About Embedded EDQ Access

Embedded EDQ access allows you to access and use Oracle Enterprise Data Quality (EDQ) within Oracle Customer Hub (UCM).

In UCM, the EDQ Launchpad allows you to download and launch Enterprise Data Quality user applications and to administer the Enterprise Data Quality server.

For more information about Oracle EDQ, see *Data Quality Guide for Oracle Customer Hub* .

Setting Up Embedded EDQ Access

To set up embedded EDQ access in Oracle Customer Hub (UCM), you must specify the symbolic URL.

To set up embedded EDQ access

1. Navigate to the Administration - Integration screen, then the WI Symbolic URL List and the Symbolic URL Administration view.
2. Find the Symbolic URL with Name field value = "UCM EDQ Access Applet."
3. Change the URL field value from `"<CHANGEME to correct EDQ Server pathname>"` to the actual EDQ server pathname.

   For example, you might change it to http://slc04ovr.us.oracle.com:9002/dndirector, if this is the pathname of your EDQ server.

Using Embedded EDQ Access

The following procedure describes how to use embedded EDQ access.

To use embedded EDQ access

1. Navigate to Administration - Universal Customer Master screen, then to Data Governance Manager and EDQ Access View.
2. Use the embedded EDQ Launchpad to download and launch EDQ user applications or to administer the EDQ server."

For more information about using Oracle EDQ, see *Data Quality Guide for Oracle Customer Hub* .

20 Oracle Customer Hub (UCM) and Data Governance Manager Web Services Reference

Oracle Customer Hub (UCM) and Data Governance Manager Web Services Reference

This chapter provides reference information for Oracle Customer Hub (UCM) and Oracle Data Governance Manager Web services. It includes the following topics:

- *Core Oracle Customer Hub (UCM) Web Services*
- *Organization Service*
- *PersonService*
- *GroupService*
- *Financial Account Service*
- *Organization Match Service*
- *PersonMatchService*
- *OrganizationCrossReferenceService*
- *PersonCrossReferenceService*
- *GroupCrossReferenceService*
- *FinancialAssetCrossReferenceService*
- *Overview of Siebel Outbound Web Services*
- *Configuring Publish and Subscribe*
- *Enabling or Disabling Oracle Customer Hub (UCM) Services*
- *About Merge Publish Service*
- *Verifying Merge Publish Settings*
- *Overview of Oracle Customer Hub (UCM) Data Enrichment Web Services*
- *Process of Configuring Data Enrichment*
- *Third-Party Data Enrichment Fields*
- *Oracle Data Governance Manager Web Services*
- *SystemsRegistration Service*
- *ConsolidateListImportService*
- *MasterService*
- *ConsolidateService*
- *ShareService*
- *CleanseCompletenessService*
- *GovernService*

Core Oracle Customer Hub (UCM) Web Services

The following Web services are used to execute core Oracle Customer Hub (UCM) operations. These Web services include:

- *Organization Service*
- *PersonService*
- *GroupService*
- *Financial Account Service*
- *Organization Match Service*
- *PersonMatchService*
- *OrganizationCrossReferenceService*
- *PersonCrossReferenceService*
- *GroupCrossReferenceService*
- *FinancialAssetCrossReferenceService*

OrganizationService

Use this Web service to perform all operations related to the organization and or the account object. This Web service is used for all party operations. Oracle Customer Hub (UCM) organization messages are used in insert, query, update and delete operations to maintain organization information.

OrganizationService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
createOrganization	Creates an organization record.
updateOrganization	Updates an existing organization record.
deleteOrganization	Deletes an organization record.
getOrganization	Retrieves organization records, based on a user query.

Request Message Description: createOrganization

For a description of this request message, see the following table.

Node	Description	Type
SwiOrganizationIO	Required. An instance of the integration object, SwiOrganizationIO. Account is the parent header and the nodes which follow in this table are considered child records.	Integration Object
Id	Row ID of the record in the requesting system cross-reference ID.	String
AccountStatus	Status of the account.	String
AccountValue	Value of the account.	String
CurrencyCode	Currency code being used by the account.	String
DUNSNumber	Dun and Bradstreet data universal numbering system. This value is a number.	String
IntegrationId	Integration ID.	String
Location	Account location.	String
MainPhoneNumber	Phone contact of the account.	String
Name	Account name.	String
NumberofEmployees	Number of employees in the organization.	String
PartyTypeCode	Party type code.	String
PartyUId	Party unique ID.	String
Status	Status.	string
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID.	String
ListOfAccount_BusinessAddress	Optional. Account business address.	Integration Component
ListOfFincorpDepositAccount	Optional. Financial corporation deposit account.	Integration Component
ListOfFincorpLoanAccount	Optional. Financial corporation account loan details.	Integration Component
ListOfContact	Optional. Contact details of the account.	Integration Component

Node	Description	Type
ListOfUcmAccountPrivacy	Optional. Privacy details of the account.	Integration Component
ListOfRelatedSalesRep	Optional. Details of related sales representatives.	Integration Component
ListOfSubAccount	Optional. Details of subordinate accounts associated with the parent account, if any.	Integration Component
ListOfRelatedOrganization	Optional. Details of the related organization.	Integration Component
ListOfRelatedIndustry	Optional. Details of related industries to the account.	Integration Component

Response Message Description: createOrganization

For a description of this response message, seethe following table.

Node	Description	Type
SwiOrganizationIO	An instance of the integration object, SwiOrganizationIO.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any.	String

Request Message Description: updateOrganization

For a description of this request message, see the following table.

Node	Description	Type
SwiOrganizationIO	Required. An instance of the integration object, SwiOrganizationIO. Account is the parent header and the nodes which follow in this table are considered child records.	Integration Object
Id	Row ID of the record in the requesting system cross-reference ID.	String
AccountStatus	Status of the account.	String

Node	Description	Type
AccountValue	Value of the account.	String
CurrencyCode	Currency code being used by the account.	String
DUNSNumber	Dun and Bradstreet data universal numbering system, number.	String
IntegrationId	Integration ID.	String
Location	Account location.	String
MainPhoneNumber	Phone contact of the account.	String
Name	Account name.	String
NumberofEmployees	Number of employees in the organization.	String
PartyTypeCode	Party type code.	String
PartyUId	Party unique ID.	String
Status	Status.	string
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID.	String
ListOfAccount_BusinessAddress	Optional. Account business address.	Integration Component
ListOfFincorpDepositAccount	Optional. Financial corporation deposit account.	Integration Component
ListOfFincorpLoanAccount	Optional. Financial corporation account loan details.	Integration Component
ListOfContact	Optional. Contact details of the account.	Integration Component
ListOfUcmAccountPrivacy	Optional. Privacy details of the account.	Integration Component
ListOfRelatedSalesRep	Optional. Details of related sales representatives.	Integration Component

Node	Description	Type
ListOfSubAccount	Optional. Details of subordinate accounts associated with the parent account, if any.	Integration Component
ListOfRelatedOrganization	Optional. Details of the related organization.	Integration Component
ListOfRelatedIndustry	Optional. Details of related industries to the account.	Integration Component

Response Message Description: updateOrganization

For a description of this response message, see the following table.

Node	Description	Type
SwiOrganizationIO	An instance of the integration object, SwiOrganizationIO, with the list of operated records.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any.	String

Request Message Description: deleteOrganization

For a description of this request message, see the following table.

Node	Description	Type
SwiOrganizationIO	Required. An instance of the integration object, SwiOrganizationIO. Account is the parent header and the nodes which follow in this table are considered child records.	Integration Object
Id	Row ID of the record in the requesting system cross-reference ID.	String
AccountStatus	Status of the account.	String
AccountValue	Value of the account.	String
CurrencyCode	Currency code being used by the account.	String

Node	Description	Type
DUNSNumber	Dun and Bradstreet data universal numbering system, number.	String
IntegrationId	Integration ID.	String
Location	Account location.	String
MainPhoneNumber	Phone contact of the account.	String
Name	Account name.	String
NumberofEmployees	Number of employees in the organization.	String
PartyTypeCode	Party type code.	String
PartyUId	Party unique ID.	String
Status	Status.	string
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID.	String
ListOfAccount_BusinessAddress	Optional. Account business address.	Integration Component
ListOfFincorpDepositAccount	Optional. Fincorp deposit account.	Integration Component
ListOfFincorpLoanAccount	Optional. Fincorp account loan details.	Integration Component
ListOfContact	Optional. Contact details of the account.	Integration Component
ListOfUcmAccountPrivacy	Optional. Privacy details of the account.	Integration Component
ListOfRelatedSalesRep	Optional. Details of related sales representatives.	Integration Component
ListOfSubAccount	Optional. Details of subordinate accounts associated with the parent account, if any.	Integration Component

Node	Description	Type
ListOfRelatedOrganization	Optional. Details of the related organization.	Integration Component
ListOfRelatedIndustry	Optional. Details of related industries to the account.	Integration Component

Response Message Description: deleteOrganization

For a description of this response message, see the following table.

Node	Description	Type
SwiOrganizationIO	The same request instance of the integration object, SwiOrganizationIO.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any	String

Request Message Description: getOrganization

For a description of this request message, see the following table.

Node	Description	Type
SwiOrganizationIO	Required. An instance of the integration object, SwiOrganizationIO. Account is the parent header and the following are the child.	Integration Object
Id	Row ID of the record in the requesting system cross-reference ID.	String
AccountStatus	Status of the account.	String
AccountValue	Value of the account.	String
CurrencyCode	Currency code being used by the account.	String
DUNSNumber	Dun and Bradstreet data universal numbering system, number.	String

Node	Description	Type
IntegrationId	Integration ID.	String
Location	Account location.	String
MainPhoneNumber	Phone contact of the account.	String
Name	Account name.	String
NumberofEmployees	Number of employees in the organization.	String
PartyTypeCode	Party type code.	String
PartyUId	Party unique ID.	String
Status	Status.	string
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID.	String
ListOfAccount_BusinessAddress	Optional. Account business address.	Integration Component
ListOfFincorpDepositAccount	Optional. Fincorp deposit account.	Integration Component
ListOfFincorpLoanAccount	Optional. Fincorp account loan details.	Integration Component
ListOfContact	Optional. Contact details of the account.	Integration Component
ListOfUcmAccountPrivacy	Optional. Privacy details of the account.	Integration Component
ListOfRelatedSalesRep	Optional. Details of related sales representatives.	Integration Component
ListOfSubAccount	Optional. Details of subordinate accounts associated with the parent account, if any.	Integration Component
ListOfRelatedOrganization	Optional. Details of the related organization.	Integration Component

Node	Description	Type
ListOfRelatedIndustry	Optional. Details of related industries to the account.	Integration Component

Response Message Description: getOrganization

For a description of this response message, see the following table.

Node	Description	Type
SwiOrganizationIO	An instance of the integration object, SwiOrganizationIO.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any.	String

OrganizationService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
OrganizationService	Workflow: UCM Organization Customer Profile Integration SOAP Process. The process properties for this workflow are as follows: • **RealtimePubSub.** The default value is True. • **Turn On CDM Cleanse.** The default value is False. • **Turn on CDM Exact Match.** The default value is False. • **Turn on CDM Match.** The default value is False. • **Turn on Survivorship.** The default value is False.	Not applicable

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
SwiOrganizationIO	Account

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
createOrganization	RunProcess
updateOrganization	RunProcess
deleteOrganization	RunProcess
getOrganization	RunProcess

PersonService

Use this Web service to create, update, retrieve and delete Person records.

PersonService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
createPerson	Creates a person record.
updatePerson	Updates an existing person record.
deletePerson	Deletes a person record.
getPerson	Retrieves person records based on a user query.

Request Message Description: createPerson

For a description of this request message, see the following table.

Node	Description	Type
SwiPersonIO	Required. An instance of the integration object, SwiPersonIO. Contact is the parent header and the following are the child.	Integration Object
The following nodes apply to SwiPersonIO		
Id	Row ID of the record in the requesting system cross-reference ID.	String
CellularPhone	Contact's mobile phone number.	String
BirthDate	Contact's birth date.	String
EmailAddress	Contact's email address.	String
FaxPhone	Contact's fax number.	String
FirstName	Contact's first name.	String
HomePhone	Contact's home phone number.	String
Income	Contact's income.	String
IncomeCurrencyCode	Income currency code.	String
IntegrationId	Integration ID.	String
LastName	Contact's last name.	String
MF	Male or female.	String
MM	Mr. or Ms.	String
ManagerIntegrationId	Manager's integration ID.	String
MaritalStatus	Contact's marital status.	String
MiddleName	Contact's middle name.	String

Node	Description	Type
MotherMaidenName	Contact's mothers' maiden name	String
PartyUId	Party unique ID.	String
PersonUId	Person user ID.	String
PreferredCommunications	Contact's preferred communication method.	String
PreferredLanguageCode	Contact preferred language code.	String
SocialSecurityNumber	Contact's social security number.	String
Status	Status.	String
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID.	String
ListOfContact_Account	Optional. Account associated with the contact.	Integration Component
ListOfUcmContactPrivacy	Optional. Contact's privacy details.	Integration Component
ListOfContact_CommunicationAddress	Optional. Contact's communication address.	Integration Component
ListOfContact_AlternatePhone	Optional. Contact's alternate phone details.	Integration Component
ListOfContact_Position	Optional. Contact's positions.	Integration Component
ListOfContact_Households	Optional. Contact's household.	Integration Component
ListOfContact_INSPersonalAddress	Optional. Contact's personal address details.	Integration Component
ListOfContact_Organization	Optional. Contact's organization details.	Integration Component
ListOfFINCORPAccount	Optional. Details of the associated FINCORP account.	Integration Component

Node	Description	Type
ListOfUCMHEConstituentName	Optional. Details of the constituent's names.	Integration Component
ListOfUCMHEConstituentIdentification	Optional. Details of the constituent's identification.	Integration Component
ListOfUCMHEConstituentAddress	Optional. Details of the constituent's addresses.	Integration Component
ListOfUCMHEConstituentAffiliation	Optional. Details of the constituent's affiliations.	Integration Component

Response Message Description: createPerson

For a description of this response message, see the following table.

Node	Description	Type
SwiPersonIO	An instance of the integration object, SwiPersonIO. Contact is the parent header.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any.	String

Request Message Description: updatePerson

For a description of this request message, see the following table.

Node	Description	Type
SwiPersonIO	Required. An instance of the integration object, SwiPersonIO. Contact is the parent header and the following are the child.	Integration Object
The following nodes apply to SwiOrganizationIO		
Id	Row ID of the record in the requesting system cross-reference ID.	String
CellularPhone	Contact's mobile phone number.	String

Node	Description	Type
BirthDate	Contact's birth date.	String
EmailAddress	Contact's email address.	String
FaxPhone	Contact's fax number.	String
FirstName	Contact's first name.	String
HomePhone	Contact's home phone number.	String
Income	Contact's income.	String
IncomeCurrencyCode	Income currency code.	String
IntegrationId	Integration ID.	String
LastName	Contact's last name.	String
MF	Male or female.	String
MM	Mr. or Ms.	String
ManagerIntegrationId	Manager's integration ID.	String
MaritalStatus	Contact's marital status.	String
MiddleName	Contact's middle name.	String
MotherMaidenName	Contact's mothers' maiden name	String
PartyUId	Party unique ID.	String
PersonUId	Person user ID.	String
PreferredCommunications	Contact's preferred communication method.	String
PreferredLanguageCode	Contact preferred language code.	String
SocialSecurityNumber	Contact's social security number.	String
Status	Status.	String

Node	Description	Type
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID.	String
ListOfContact_Account	Optional. Account associated with the contact.	Integration component
ListOfUcmContactPrivacy	Optional. Contact's privacy details.	Integration component
ListOfContact_CommunicationAddress	Optional. Contact's communication address.	Integration component
ListOfContact_AlternatePhone	Optional. Contact's alternate phone details.	Integration component
ListOfContact_Position	Optional. Contact's positions.	Integration component
ListOfContact_Households	Optional. Contact's household.	Integration component
ListOfContact_INSPersonalAddress	Optional. Contact's personal address details.	Integration component
ListOfContact_Organization	Optional. Contact's organization details.	Integration component
ListOfFINCORPAccount	Optional. Details of the associated FINCORP account.	Integration component
ListOfUCMHEConstituentName	Optional. Details of the constituent's names.	Integration component
ListOfUCMHEConstituentIdentification	Optional. Details of the constituent's identification.	Integration component
ListOfUCMHEConstituentAddress	Optional. Details of the constituent's addresses.	Integration component
ListOfUCMHEConstituentAffiliation	Optional. Details of the constituent's affiliations.	Integration component

Response Message Description: updatePerson

For a description of this response message, see the following table.

Node	Description	Type
SwiPersonIO	An instance of the integration object, SwiPersonIO. Contact is the parent header.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any.	String

Request Message Description: deletePerson

For a description of this request message, see the following table.

Node	Description	Type
SwiPersonIO	Required. An instance of the integration object, SwiPersonIO. Contact is the parent header and the following are the child.	Integration Object
The following nodes apply to SwiOrganizationIO		
Id	Row ID of the record in the requesting system cross-reference ID.	String
CellularPhone	Contact's mobile phone number.	String
BirthDate	Contact's birth date.	String
EmailAddress	Contact's email address.	String
FaxPhone	Contact's fax number.	String
FirstName	Contact's first name.	String
HomePhone	Contact's home phone number.	String
Income	Contact's income.	String
IncomeCurrencyCode	Income currency code.	String
IntegrationId	Integration ID.	String

Node	Description	Type
LastName	Contact's last name.	String
MF	Male or female.	String
MM	Mr. or Ms.	String
ManagerIntegrationId	Manager's integration ID.	String
MaritalStatus	Contact's marital status.	String
MiddleName	Contact's middle name.	String
MotherMaidenName	Contact's mothers' maiden name	String
PartyUId	Party unique ID.	String
PersonUId	Person user ID.	String
PreferredCommunications	Contact's preferred communication method.	String
PreferredLanguageCode	Contact preferred language code.	String
SocialSecurityNumber	Contact's social security number.	String
Status	Status.	String
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID.	String
ListOfContact_Account	Optional. Account associated with the contact.	Integration Component
ListOfUcmContactPrivacy	Optional. Contact's privacy details.	Integration Component
ListOfContact_CommunicationAddress	Optional. Contact's communication address.	Integration Component
ListOfContact_AlternatePhone	Optional. Contact's alternate phone details.	Integration Component
ListOfContact_Position	Optional. Contact's positions.	Integration Component

Node	Description	Type
ListOfContact_Households	Optional. Contact's household.	Integration Component
ListOfContact_INSPersonalAddress	Optional. Contact's personal address details.	Integration Component
ListOfContact_Organization	Optional. Contact's organization details.	Integration Component
ListOfFINCORPAccount	Optional. Details of the associated FINCORP account.	Integration Component
ListOfUCMHEConstituentName	Optional. Details of the constituent's names.	Integration Component
ListOfUCMHEConstituentIdentification	Optional. Details of the constituent's identification.	Integration Component
ListOfUCMHEConstituentAddress	Optional. Details of the constituent's addresses.	Integration Component
ListOfUCMHEConstituentAffiliation	Optional. Details of the constituent's affiliations.	Integration Component

Response Message Description: deletePerson

For a description of this response message, see the following table.

Node	Description	Type
SwiPersonIO	Required. The same request instance of the integration object, SwiPersonIO.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any	String

Request Message Description: getPerson

For a description of this request message, see the following table.

Node	Description	Type
SwiPersonIO	Required. An instance of the integration object, SwiPersonIO. Contact is the parent header and the following are the child.	Integration Object
The following nodes apply to SwiOrganizationIO		
Id	Row ID of the record in the requesting system cross-reference ID.	String
CellularPhone	Contact's mobile phone number.	String
BirthDate	Contact's birth date.	String
EmailAddress	Contact's email address.	String
FaxPhone	Contact's fax number.	String
FirstName	Contact's first name.	String
HomePhone	Contact's home phone number.	String
Income	Contact's income.	String
IncomeCurrencyCode	Income currency code.	String
IntegrationId	Integration ID.	String
LastName	Contact's last name.	String
MF	Male or female.	String
MM	Mr. or Ms.	String
ManagerIntegrationId	Manager's integration ID.	String
MaritalStatus	Contact's marital status.	String
MiddleName	Contact's middle name.	String
MotherMaidenName	Contact's mothers' maiden name	String
PartyUId	Party unique ID.	String

Node	Description	Type
PersonUId	Person user ID.	String
PreferredCommunications	Contact's preferred communication method.	String
PreferredLanguageCode	Contact preferred language code.	String
SocialSecurityNumber	Contact's social security number.	String
Status	Status.	String
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID.	String
ListOfContact_Account	Optional. Account associated with the contact.	Integration Component
ListOfUcmContactPrivacy	Optional. Contact's privacy details.	Integration Component
ListOfContact_CommunicationAddress	Optional. Contact's communication address.	Integration Component
ListOfContact_AlternatePhone	Optional. Contact's alternate phone details.	Integration Component
ListOfContact_Position	Optional. Contact's positions.	Integration Component
ListOfContact_Households	Optional. Contact's household.	Integration Component
ListOfContact_INSPersonalAddress	Optional. Contact's personal address details.	Integration Component
ListOfContact_Organization	Optional. Contact's organization details.	Integration Component
ListOfFINCORPAccount	Optional. Details of the associated FINCORP account.	Integration Component
ListOfUCMHEConstituentName	Optional. Details of the constituent's names.	Integration Component
ListOfUCMHEConstituentIdentification	Optional. Details of the constituent's identification.	Integration Component

Node	Description	Type
ListOfUCMHEConstituentAddress	Optional. Details of the constituent's addresses.	Integration Component
ListOfUCMHEConstituentAffiliation	Optional. Details of the constituent's affiliations.	Integration Component

Response Message Description: getPerson

For a description of this response message, see the following table.

Node	Description	Type
SwiPersonIO	An instance of the integration object, SwiPersonIO. Contact is the parent header.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any.	String

PersonService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
PersonService	Workflow: UCM Person Customer Profile Integration SOAP Process. The process properties for this workflow are as follows: • RealtimePubSub. The default value is True. • Turn On CDM Cleanse. The default value is False. • Turn on CDM Exact Match. The default value is False.	Not applicable

Siebel Repository Name	Boundary Object Type	Class
	• Turn on CDM Match. The default value is False. • Turn on Survivorship. The default value is False.	

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
SwiPersonIO	Contact

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
createPerson	RunProcess
updatePerson	RunProcess
deletePerson	RunProcess
getPerson	RunProcess

GroupService

Use this Web service to perform all operations related to Households (Groups). UCM Household messages are used in insert, query, update and delete operations to maintain Household information.

GroupService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
createGroup	Creates a new group record.
updateGroup	Updates and existing group record.

Name	Description
deleteGroup	Deletes a group record.
getGroup	Retrieves group records, based on a user query.

Request Message Description: createGroup

For a description of this request message, see the following table.

Node	Description	Type
SwiGroupIO	Required. An instance of the integration object, SwiGroupIO. Household is the parent header and the following are the child	Integration Object
The following nodes apply to SwiGroupIO		
Id	Row ID of the record in the requesting system cross-reference ID.	String
HouseholdName	Name of the household.	String
HouseholdNumber	Household number.	String
IntegrationId	Integration ID.	String
Name	Household name.	String
PartyUId	Party unique ID.	String
PhoneNumber	Phone number.	String
PrivacyCode	Privacy code	String
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID	String
HouseholdLink	Household link	String
LengthOfResidence	Length of residence	String

Node	Description	Type
HomeOwnerRenter	Rental home owner	String
ListOfHousehold_HouseholdAddress	Optional. Household address.	Integration Component
ListOfHousehold_Position	Optional. Positions of the household	Integration Component
ListOfUCMFINCORPAccountPrivacy	Optional. Privacy details of the FINCORP account	Integration Component
ListOfContact	Optional. Contact details of the household	Integration Component
ListOfHousehold_Organization	Optional. Organization details related to the household.	Integration Component

Response Message Description: createGroup

For a description of this response message, see the following table.

Node	Description	Type
SwiGroupIO	An instance of the integration object, SwiGroupIO, with the list of records operated. Household is the parent header.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any.	String

Request Message Description: updateGroup

For a description of this request message, see the following table.

Node	Description	Type
SwiGroupIO	Required. An instance of the integration object, SwiGroupIO. Household is the parent header and the following are the child	Integration Object
The following nodes apply to SwiGroupIO		

Node	Description	Type
Id	Row ID of the record in the requesting system cross-reference ID.	String
HouseholdName	Name of the household.	String
HouseholdNumber	Household number.	String
IntegrationId	Integration ID.	String
Name	Household name.	String
PartyUId	Party unique ID.	String
PhoneNumber	Phone number.	String
PrivacyCode	Privacy code	String
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID	String
HouseholdLink	Household link	String
LengthOfResidence	Length of residence	String
HomeOwnerRenter	Rental home owner	String
ListOfHousehold_HouseholdAddress	Optional. Household address.	Integration Component
ListOfHousehold_Position	Optional. Positions of the household.	Integration Component
ListOfUCMFINCORPAccountPrivacy	Optional. Privacy details of the FINCORP account	Integration Component
ListOfContact	Optional. Contact details of the household.	Integration Component
ListOfHousehold_Organization	Optional. Organization details related to the household.	Integration Component

Response Message Description: updateGroup

For a description of this response message, see the following table.

Node	Description	Type
SwiGroupIO	An instance of the integration object, SwiGroupIO, with the list of records operated. Household is the parent header.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any.	String

Request Message Description: deleteGroup

For a description of this request message, see the following table.

Node	Description	Type
SwiGroupIO	Required. An instance of the integration object, SwiGroupIO. Household is the parent header and the following are the child	Integration Object
The following nodes apply to SwiGroupIO		
Id	Row ID of the record in the requesting system cross-reference ID.	String
HouseholdName	Name of the household.	String
HouseholdNumber	Household number.	String
IntegrationId	Integration ID.	String
Name	Household name.	String
PartyUId	Party unique ID.	String
PhoneNumber	Phone number.	String
PrivacyCode	Privacy code	String

Node	Description	Type
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID.	String
HouseholdLink	Household link	String
LengthOfResidence	Length of residence	String
HomeOwnerRenter	Rental home owner	String
ListOfHousehold_HouseholdAddress	Optional. Household address.	Integration Component
ListOfHousehold_Position	Optional. Positions of the household.	Integration Component
ListOfUCMFINCORPAccountPrivacy	Optional. Privacy details of the FINCORP account	Integration Component
ListOfContact	Optional. Contact details of the household.	Integration Component
ListOfHousehold_Organization	Optional. Organization details related to the household.	Integration Component

Response Message Description: deleteGroup

For a description of this response message, see the following table.

Node	Description	Type
SwiGroupIO	Required. The same request instance of the integration object, SwiGroupIO.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any	String

Request Message Description: getGroup

For a description of this request message, see the following table.

Node	Description	Type
SwiGroupIO	Required. An instance of the integration object, SwiGroupIO. Household is the parent header and the following are the child	Integration Object
The following nodes apply to SwiGroupIO		
Id	Row ID of the record in the requesting system cross-reference ID.	String
HouseholdName	Name of the household.	String
HouseholdNumber	Household number.	String
IntegrationId	Integration ID.	String
Name	Household name.	String
PartyUId	Party unique ID.	String
PhoneNumber	Phone number.	String
PrivacyCode	Privacy code	String
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID	String
HouseholdLink	Household link	String
LengthOfResidence	Length of residence	String
HomeOwnerRenter	Rental home owner	String
ListOfHousehold_HouseholdAddress	Optional. Household address	Integration Component
ListOfHousehold_Position	Optional. Positions of the household.	Integration Component
ListOfUCMFINCORPAccountPrivacy	Optional. Privacy details of the FINCORP account.	Integration Component
ListOfContact	Optional. Contact details of the household.	Integration Component

Node	Description	Type
ListOfHousehold_Organization	Optional. Organization details related to the household.	Integration Component

Response Message Description: getGroup

For a description of this response message, see the following table.

Node	Description	Type
SwiGroupIO	An instance of the integration object, SwiGroupIO with the list of records operated. Household is the parent header.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any.	String

GroupService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
GroupService	Workflow: UCM Group Customer Profile Integration SOAP Process. The process properties for this workflow are as follows: • RealtimePubSub. The default value is True. • Turn On CDM Cleanse. The default value is False. • Turn on CDM Exact Match. The default value is False. • Turn on CDM Match. The default value is False.	Not applicable

Siebel Repository Name	Boundary Object Type	Class
	• Turn on Survivorship. The default value is False.	

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
SwiGroupIO	Household

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
createGroup	RunProcess
updateGroup	RunProcess
deleteGroup	RunProcess
getGroup	RunProcess

Financial Account Service

Use this Web service to create new financial service account records and to perform update, delete and retrieve operations.

Financial Account Service Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
createFinancialAsset	Creates a new financial account record.
updateFinancialAsset	Updates an existing financial account record.

Name	Description
deleteFinancialAsset	Deletes a financial account record.
getFinancialAsset	Retrieves financial account records, based on a user query.

Request Message Description: createFinancialAsset

For a description of this request message, see the following table.

Node	Description	Type
SwiFinancialAssetIO	Required. An instance of the integration object, SwiFinancialAssetIO. FINCORP. Account is the parent header and the nodes which follow in this table are considered child records.	Integration Object
The following nodes apply to SwiFinancialAssetIO		
Id	Row ID of the record in the requesting system cross-reference ID.	String
AccountName	Account name.	String
AccountNumber	Number of the account.	String
IntegrationId	Integration ID.	String
Name	FINCORP account name.	String
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID	String
ListOfCutAddress	Optional. Business address.	Integration Component
ListOfFINCORPAccountContact	Optional. Contacts of the FINCORP account.	Integration Component
ListOfUCMFINCORPAccountPrivacy	Optional. Privacy details of the FINCORP account.	Integration Component
ListOfFinsCfFinancialAssetAddress	Optional. Address details of the FINCORP account.	Integration Component

Node	Description	Type
ListOfFINCORPAccountContact	Optional. Contact details of the FINCORP account.	Integration Component
ListOfContact	Contact details.	Integration Component

Response Message Description: createFinancialAsset

For a description of this response message, see the following table.

Node	Description	Type
SwiFinancialAssetIO	An instance of the integration object, SwiFinancialAssetIO. Financial corporation. Account is the parent header.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any.	String

Request Message Description: updateFinancialAsset

For a description of this request message, see the following table.

Node	Description	Type
SwiFinancialAssetIO	Required. An instance of the integration object, SwiFinancialAssetIO. FINCORP account is the parent header and the following are child nodes.	Integration Object
The following nodes apply to SwiFinancialAssetIO		
Id	Row ID of the record in the requesting system cross-reference ID.	String
AccountName	Account name.	String
AccountNumber	Account number.	String
IntegrationId	Integration ID.	String

Node	Description	Type
Name	FINCORP account name.	String
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID.	String
ListOfCutAddress	Optional. Business address.	Integration Component
ListOfFINCORPAccountContact	Optional. Contacts of the FINCORP account.	Integration Component
ListOfUCMFINCORPAccountPrivacy	Optional. Privacy details of the FINCORP account.	Integration Component
ListOfFinsCfFinancialAssetAddress	Optional. Address details of the FINCORP account.	Integration Component
ListOfFINCORPAccountContact	Optional. Contact details of the FINCORP account.	Integration Component
ListOfContact	Contact details.	Integration Component

Response Message Description: updateFinancialAsset

For a description of this response message, see the following table.

Node	Description	Type
SwiFinancialAssetIO	An instance of the integration object, SwiFinancialAssetIO. FINCORP Account is the parent header	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any.	String

Request Message Description: deleteFinancialAsset

For a description of this request message, see the following table.

Node	Description	Type
SwiFinancialAssetIO	Required. An instance of the integration object, SwiFinancialAssetIO. FINCORP. Account is the parent header and the following are child nodes.	Integration Object
The following nodes apply to SwiFinancialAssetIO		
Id	Row ID of the Record in the requesting system cross-reference ID.	String
AccountName	Account Name	String
AccountNumber	Number of the Account	String
IntegrationId	Integration ID	String
Name	FINCORP Account Name	String
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID	String
ListOfCutAddress	Optional. Business address	Integration Component
ListOfFINCORPAccountContact	Optional. Contacts of the FINCORP Account.	Integration Component
ListOfUCMFINCORPAccountPrivacy	Optional. Privacy details of the FINCORP Account	Integration Component
ListOfFinsCfFinancialAssetAddress	Optional. Address details of the FINCORP Account	Integration Component
ListOfFINCORPAccountContact	Optional. Contact details of the FINCORP Account	Integration Component
ListOfContact	Contact details.	Integration Component

Response Message Description: deleteFinancialAsset

For a description of this response message, see the following table.

Node	Description	Type
SwiFinancialAssetIO	Required. The same request instance of the integration object, SwiFinancialAssetIO.	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any	String

Request Message Description: getFinancialAsset

For a description of this request message, see the following table .

Node	Description	Type
SwiFinancialAssetIO	Required. An instance of the integration object, SwiFinancialAssetIO. FINCORP Account is the parent header and the following are child nodes.	Integration Object
The following nodes apply to SwiFinancialAssetIO		
Id	Row ID of the record in the requesting system cross-reference ID.	String
AccountName	Account Name	String
AccountNumber	Number of the Account	String
IntegrationId	Integration ID	String
Name	FINCORP Account Name	String
AIAUCMIntegrationId	Oracle Application Integration Architecture and Oracle Customer Hub (UCM) integration ID	String
ListOfCutAddress	Optional. Business address	Integration Component
ListOfFINCORPAccountContact	Optional. Contacts of the FINCORP Account.	Integration Component
ListOfUCMFINCORPAccountPrivacy	Optional. Privacy details of the FINCORP Account	Integration Component

Node	Description	Type
ListOfFinsCfFinancialAssetAddress	Optional. Address details of the FINCORP Account	Integration Component
ListOfFINCORPAccountContact	Optional. Contact details of the FINCORP Account	Integration Component
ListOfContact	Contact details.	Integration Component

Response Message Description: getFinancialAsset

For a description of this response message, see the following table.

Node	Description	Type
SwiFinancialAssetIO	An instance of the integration object, SwiFinancialAssetIO. FINCORP. Account is the parent header	Integration Object
Error Code	Error code, if any.	String
Error Message	Error message, if any.	String
Error Symbol	Error symbol, if any.	String

FinancialAssetService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
FinancialAssetService	Workflow: UCM Financial Asset Customer Profile Integration SOAP Process. The process properties for this workflow are as follows: • RealtimePubSub. The default value is True. • Turn On CDM Cleanse. The default value is False.	Not applicable

Siebel Repository Name	Boundary Object Type	Class
	• Turn on CDM Exact Match. The default value is False. • Turn on CDM Match. The default value is False. • Turn on Survivorship. The default value is False.	

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
SwiFinancialAssetIO	FINCORP Account

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
createFinancialAsset	RunProcess
updateFinancialAsset	RunProcess
deleteFinancialAsset	RunProcess
getFinancialAsset	RunProcess

OrganizationMatchService

Use this Web service to find similar accounts and organizations in designated databases, such as Oracle Customer Hub (UCM) database, corresponding to the one being sent in a request message. The similarity is based on advanced algorithms, as opposed to a plain search where a user would input A* and receive all Account records that begin with A. Additionally the algorithm depends on the matching engine (such as ISS, EDQ and so on) used at the back-end with the configured match strategies, rules, and configurations.

OrganizationMatchService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
OrganizationMatch	Locates similar accounts, based on predefined rules and algorithms.

Request Message Description: OrganizationMatch

For a description of this request message, see the following table.

Node	Description	Type
SwiOrganizationMatchInputIO	An instance of the integration object, SwiOrganizationMatchInputIO	Integration Object

Response Message Description: OrganizationMatch

For a description of this response message, see the following table.

Node	Description	Type
SwiOrganizationMatchOutputIO	An instance of the integration object, SwiOrganizationMatchOutputIO	Integration Object

OrganizationMatchService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
DeDuplication	Business Service	CSSDeDupService

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
SwiOrganizationMatchInputIO	SwiOrganizationMatchInputIO

Siebel Repository Name	External Name
SwiOrganizationMatchOutputIO	SwiOrganizationMatchOutputIO

Note: When adding additional fields to the ICs in SwiOrganizationMatchInputIO and SwiOrganizationMatchOutputIO, the IC Field Name should not have any spaces in between. The External Name should match with the BC Field Name, which may have spaces in between.

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
OrganizationMatch	Organization Match Web service

PersonMatchService

Use this Web service to find similar contacts in designated databases, such as Oracle Customer Hub (UCM) database, corresponding to the one being sent in a request message. The similarity is based on an advanced algorithms (as opposed to a plain search where a user would input A* and receive all Account records beginning with A) and depends on the matching engine (such as ISS, EDQ and so on) used at the back-end with the match strategies, rules, and configurations in place.

PersonMatchService Operations

For the operation associated with this Web service, see the following.

Name	Description
PersonMatch	Finds similar contacts, based on predefined rules and algorithms.

Request Message Description: PersonMatch

For a description of this request message, see the following table.

Node	Description	Type
SwiPersonMatchInputIO	An instance of the SwiPersonMatchInputIO Integration Object.	Integration Object

Response Message Description: PersonMatch

For a description of this response message, see the following table.

Node	Description	Type
SwiPersonMatchOutputIO	An instance of the SwiPersonMatchOutputIO Integration Object.	Integration Object

PersonMatchService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
DeDuplication	Business Service	CSSDeDupService

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
SwiPersonMatchInputIO	SwiPersonMatchInputIO
SwiPersonMatchOutputIO	SwiPersonMatchOutputIO

Note: When adding additional fields to the ICs in SwiPersonMatchInputIO and SwiPersonMatchOutputIO, the IC Field Name should not have any spaces in between. The External Name should match with the BC Field Name, which may have spaces in between.

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
PersonMatch	Person Match Web service

OrganizationCrossReferenceService

Use this Web service to create explicit links between organization records in Oracle Customer Hub (UCM) and those in participating applications.

OrganizationCrossReferenceService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
createOrganizationCrossReference	Creates a cross-reference ID to a Organization record.
updateOrganizationCrossReference	Updates a cross-reference ID to a Organization record.
deleteOrganizationCrossReference	Deletes a cross-reference ID to a Organization record.
getOrganizationCrossReference	Retrieves a cross-reference ID to a Organization record.

Request Message Description: createOrganizationCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiOrganizationPublish	An instance of the integration object, SwiOrganizationPublish. The main Integration components are Account and CIF Account Reference.	Integration Object

Response Message Description: createOrganizationCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiOrganizationPublish	An instance of the integration object, SwiOrganizationPublish. The main Integration components are Account and CIF Account Reference.	Integration Object

Node	Description	Type

Request Message Description: updateOrganizationCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiOrganizationPublish	An instance of the integration object, SwiOrganizationPublish. The main integration components are Account and CIF Account Reference.	Integration Object

Response Message Description: updateOrganizationCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiOrganizationPublishIO	An instance of the integration object, SwiOrganizationPublish. The main components are Account and CIF Account Reference.	Integration Object

Request Message Description: deleteOrganizationCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiOrganizationPublishIO	An instance of the integration object, SwiOrganizationPublish. The main Integration components are Account and CIF Account Reference.	Integration Object

Response Message Description: deleteOrganizationCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiOrganizationPublishIO	An instance of the integration object, SwiOrganizationPublish. The main Integration components are Account and CIF Account Reference.	Integration Object

Request Message Description: getOrganizationCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiOrganizationPublishIO	An instance of the integration object, SwiOrganizationPublish. The main component is Account.	Integration Object

Response Message Description: getOrganizationCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiOrganizationPublishIO	An instance of the integration object, SwiOrganizationPublish. The main components are are Account and CIF Account Reference.	Integration Object

OrganizationCrossReferenceService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
UCM Cross Reference Service	Business Service	CSSUCMXRefService

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
SwiOrganizationPublishIO	Account

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
createOrganizationCrossReference	CreateOrganizationCrossReference

Operation	Method
updateOrganizationCrossReference	UpdateOrganizationCrossReference
deleteOrganizationCrossReference	DeleteOrganizationCrossReference
getOrganizationCrossReference	GetOrganizationCrossReference

PersonCrossReferenceService

Use this Web service to create explicit links between person records in Oracle Customer Hub (UCM) and those in participating applications.

PersonCrossReferenceService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
createPersonCrossReference	Creates a cross-reference ID to a Person record.
updatePersonCrossReference	Updates a cross-reference ID to a Person record.
deletePersonCrossReference	Deletes a cross-reference ID to a Person record.
getPersonCrossReference	Retrieves a cross-reference ID to a Person record.

Request Message Description: createPersonCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiPersonPublishIO	An instance of the integration object, SwiPersonPublishIO. The main components are Contact and CIF Contact Reference.	Integration Object

Response Message Description: createPersonCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiPersonPublishIO	An instance of the integration object, SwiPersonPublishIO. The main components are Contact and CIF Contact Reference.	Integration Object

Request Message Description: updatePersonCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiPersonPublishIO	An instance of the integration object, SwiPersonPublishIO. The main components are Contact and CIF Contact Reference.	Integration Object

Response Message Description: updatePersonCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiPersonPublishIO	An instance of the integration object, SwiPersonPublishIO. The main components are Contact and CIF Contact Reference.	Integration Object

Request Message Description: deletePersonCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiPersonPublishIO	An instance of the integration object, SwiPersonPublishIO. The main components are Contact and CIF Contact Reference.	Integration Object

Response Message Description: deletePersonCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiPersonPublishIO	An instance of the integration object, SwiPersonPublishIO. The main components are Contact and CIF Contact Reference.	Integration Object

Request Message Description: getPersonCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiPersonPublishIO	An instance of the integration object, SwiPersonPublishIO. The main component is Contact.	Integration Object

Response Message Description: getPersonCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiPersonPublishIO	An instance of the integration object, SwiPersonPublishIO. The main components are Contact and CIF Contact Reference.	Integration Object

PersonCrossReferenceService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
UCM Cross Reference Service	Business Service	CSSUCMXRefService

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
SwiPersonPublishIO	Contact

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
createPersonCrossReference	CreatePersonCrossReference

Operation	Method
updatePersonCrossReference	UpdatePersonCrossReference
deletePersonCrossReference	DeletePersonCrossReference
getPersonCrossReference	GetPersonCrossReference

GroupCrossReferenceService

Use this Web service to create explicit links between Group records in Oracle Customer Hub (UCM) and those in participating applications.

GroupCrossReferenceService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
createGroupReference	Creates a cross-reference ID to a Group record.
updateGroupReference	Updates a cross-reference ID to a Group record.
deleteGroupReference	Deletes a cross-reference ID to a Group record.
getGroupReference	Retrieves a cross-reference ID to a Group record.

Request Message Description: createGroupCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiGroupPublishIO	An instance of the integration object SwiGroupPublishIO.	Integration Object

Response Message Description: createGroupCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiGroupPublishIO	An instance of the integration object, SwiGroupPublishIO.	Integration Object

Request Message Description: updateGroupCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiGroupPublishIO	An instance of the integration object, SwiGroupPublishIO.	Integration Object

Integration ObjectResponse Message Description: updateGroupCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiGroupPublishIO	An instance of the integration object, SwiGroupPublishIO.	Integration Object

Request Message Description: deleteGroupCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiGroupPublishIO	An instance of the integration object, SwiGroupPublishIO.	Integration Object

Response Message Description: deleteGroupCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiGroupPublishIO	An instance of the integration object, SwiGroupPublishIO.	Integration Object

Request Message Description: getGroupCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiGroupPublishIO	An instance of the integration object, SwiGroupPublishIO.	Integration Object

Node	Description	Type

Response Message Description: getGroupCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiGroupPublishIO	An instance of the integration object, SwiGroupPublishIO.	Integration Object

GroupCrossReferenceService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
UCM Cross Reference Service	Business Service	CSSUCMXRefService

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
SwiGroupPublishIO	Household

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
createGroupCrossReference	CreateGroupCrossReference
updateGroupCrossReference	UpdateGroupCrossReference

Operation	Method
deleteGroupCrossReference	DeleteGroupCrossReference
getGroupCrossReference	GetGroupCrossReference

FinancialAssetCrossReferenceService

Use this Web service to create explicit links between financial account records within Oracle Customer Hub (UCM) and those in participating applications.

FinancialAssetCrossReferenceService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
createFinancialAssetReference	Creates a cross-reference ID to a financial asset record.
updateFinancialAssetReference	Updates a cross-reference ID to a financial asset record.
deleteFinancialAssetReference	Deletes a cross-reference ID to a financial asset record.
getFinancialAssetReference	Retrieves a cross-reference ID to a financial asset record.

Request Message Description: createFinancialAssetCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiFinancialAssetPublishIO	An instance of the integration object, SwiFinancialAssetPublishIO.	Integration Object

Response Message Description: createFinancialAssetCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiFinancialAssetPublishIO	An instance of the integration object, SwiFinancialAssetPublishIO.	Integration Object

Request Message Description: updateFinancialAssetCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiFinancialAssetPublishIO	An instance of the integration object, SwiFinancialAssetPublishIO.	Integration Object

Response Message Description: updateFinancialAssetCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiFinancialAssetPublishIO	An instance of the integration object, SwiFinancialAssetPublishIO.	Integration Object

Request Message Description: deleteFinancialAssetCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiFinancialAssetPublishIO	An instance of the integration object, SwiFinancialAssetPublishIO.	Integration Object

Response Message Description: deleteFinancialAssetCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiFinancialAssetPublishIO	An instance of the integration object, SwiFinancialAssetPublishIO.	Integration Object

Request Message Description: getFinancialAssetCrossReference

For a description of this request message, see the following table.

Node	Description	Type
SwiFinancialAssetPublishIO	An instance of the integration object, SwiFinancialAssetPublishIO.	Integration Object

Response Message Description: getFinancialAssetCrossReference

For a description of this response message, see the following table.

Node	Description	Type
SwiFinancialAssetPublishIO	An instance of the integration object, SwiFinancialAssetPublishIO.	Integration Object

FinancialAssetCrossReferenceService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
UCM Cross Reference Service	Business Service	CSSUCMXRefService

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
SwiFinancialAssetPublishIO	FINCORP Account

Methods for FinancialAssetCrossReferenceService

For a description of the methods for this Web service, see the following table.

Operation	Method
createFinancialAssetCrossReference	CreateFinancialAssetCrossReference

Operation	Method
updateFinancialAssetCrossReference	UpdateFinancialAssetCrossReference
deleteFinancialAssetCrossReference	DeleteFinancialAssetCrossReference
getFinancialAssetCrossReference	GetFinancialAssetCrossReference

Overview of Siebel Outbound Web Services

Siebel Outbound services are Web services that are configured to call Oracle Application Integration Architecture services through proxy business services. The proxy business services are created in Oracle Customer Hub (UCM) by using WSDL from Oracle Application Integration Architecture.

A Siebel Outbound Web service is used to send and receive data to and from Siebel Business Applications through synchronous or asynchronous service calls for different process flows.

Configuring Publish and Subscribe

The Publish Subscribe Web service provides a means of publishing messages to a middleware system, which acts as a funnel to the subscribing applications.

Note: This configuration task is not required when integrating Oracle Application Integration Architecture with Oracle Customer Hub (UCM). A middleware installation must be completed and set up to handle outbound publish actions. If you are setting up an integration of Oracle Application Integration Architecture with Oracle Customer Hub (UCM), make sure Oracle Application Integration Architecture is set up and functioning. If Oracle Customer Hub (UCM) is integrated with any other middleware provider, make sure that the middleware installation is completed before doing the following task.

To create a proxy business service

1. In Siebel Tools, lock the project Web Service Integration, and lock the Project name.
2. From the File menu, select New Object.
3. In the New Object wizard, click the EAI tab, and click the Web Service icon.
4. Find the PublishSubscribe WSDL file in your installation directory.

   The run-time data file and log file entries are automatically created.

   CAUTION: Do not select the Process Fault Schema check box because it will produce unintended results.
5. Click Next.
6. Select the Deploy Integration Objects and Proxy Business Services check box.
7. Click Finish.

8. Search the log files for the entry, *Outbound proxy business service definitions to be generated*. At the end of the definition, you see the name of the proxy business service you created.

   For example:

   ```
   Outbound proxy business service definitions to be generated:

   SyncOrganizationUCMJMSProducerService_1 for portTypeName:http://xmlns.oracle.com/

   ABCS/UCM/Core/SyncOrganizationUCMJMSProducer/

   V1:SyncOrganizationUCMJMSProducerService
   ```

   In this example, SyncOrganizationUCMJMSProducerService is the name of the proxy service that was created.
9. When the proxy business service is created, click the Business Services object in the Object Explorer.

Modifying the Newly Created Proxy Business Service

Use the following task to modify the newly created proxy business service.

Note: This configuration task is not required when integrating Oracle Application Integration Architecture with Oracle Customer Hub (UCM).

To modify the newly created business service

1. In Siebel Tools Business Services screen, query for the newly created proxy business service.
2. In the Object Explorer, navigate to Business Service Method Arguments.
3. Search for arguments of type Integration Object.
4. Note the Integration Object names, then using the Object Explorer, find and delete the objects.

   The default integration objects will not be used. Instead, the newly imported integration objects will be used.
5. Replace the existing Integration Object names in the Business Service Method Arguments of the proxy service with the new Integration Object names displayed in the following table.

Base Object Type of the Proxy Service	New Integration Object Name
Organization	SwiOrganizationPublishIO
Person	SwiPersonPublishIO
Household	SwiGroupPublishIO
FINCORP Account	SwiFinancialAssetPublishIO

6. Compile the affected objects, then restart the Siebel Server.

Registering the Web Services

Use this task to register the newly created Web services.

To register the newly created Web services

1. From the Siebel Mobile Web client, navigate to the Administration - Web Services screen.
2. In the Deployed Integration Objects view verify that all the Integration Objects you deployed in Siebel Tools are showing the correct time of deployment.

   This will allow you to make sure the Integration Objects displayed are those you imported.

   > **Note:** If you have an integration of Oracle Application Integration Architecture with Oracle Customer Hub (UCM), go to Step 3. If you do not have an integration, go to Step 5.

3. From the Outbound Web Services view, query for the Oracle Application Integration Architecture Web services:
   - SyncOrganizationUCMJMSProducer
   - SyncPersonUCMJMSProducer
4. Change the address of the ports corresponding to these Web services to point to the appropriate Oracle Application Integration Architecture end point.
5. If you are not integrated with Oracle Application Integration Architecture, navigate to the Outbound Web Services view, and query for the Web service that was created in *Creating the Proxy Business Service*.
6. Change the address of the ports corresponding to the Web services to point to the appropriate middleware end point.

Enabling or Disabling Oracle Customer Hub (UCM) Services

After configuring the services, you must enable them. The following tasks allow you to enable or disable the Web services.

Configuring Publish/Subscribe System Preferences

You must set the Publish/Subscribe settings to allow you to use the RealTime Publish/Subscribe service. Use the following task to configure Publish/Subscribe system preferences.

To configure Publish/Subscribe system preferences

1. Navigate to the Administration - Application screen, then the System Preferences view.
2. In the System Preferences screen, query for Enable Realtime PubSub.

3. If no preferences are found, create new records for each of the parameters displayed in the following table.

Parameter	Value
Enable Realtime PubSub 1	Account
Enable Realtime PubSub 2	Contact
Enable Realtime PubSub 3	Household
Enable Realtime PubSub 4	FINCORP Account

About Merge Publish Service

If a data steward performs a merge operation, the surviving record is updated and the victim record is deleted in the best version table. The surviving record and victim record that are affected by the merge operation in Oracle Customer Hub (UCM) are published to the applications by using the publish subscribe business service.

Verifying Merge Publish Settings

There is no additional configuration needed for these settings, though you must verify the configurations. Use the following task to verify Merge Publish settings.

To verify the Merge Publish settings

1. In the Siebel Tools Object Explorer, choose Business Components.
2. In the Business Components screen, query for UCM DeDuplication Results.
3. Verify that the query has returned the following values:
    - UCM DeDuplication Results (Account)
    - UCM DeDuplication Results (Contact)
4. Select UCM DeDuplication Results (Account) and in the Object Explorer, navigate to Business Component User Prop.
5. Verify that the following values are present for the UCM DeDuplication Results (Account) business component:

User Property Name	User Property Value
MERGE_PUBLISH_IO	SwiOrganizationPublishIO

User Property Name	User Property Value
PUBLISH_METHOD_TYPE	PublishMethod
VICTIM_OPERATION_TYPE	Delete
SURVIVOR_OPERATION_TYPE	Upsert

6. Select UCM DeDuplication Results (Contact) and in the Object Explorer, navigate to Business Component User Prop.
7. Verify that the following values are present for the UCM DeDuplication Results (Contact) business component:

User Property Name	User Property Value
MERGE_PUBLISH_IO	SwiPersonPublishIO
PUBLISH_METHOD_TYPE	PublishMethod
VICTIM_OPERATION_TYPE	Delete
SURVIVOR_OPERATION_TYPE	Upsert

Overview of Oracle Customer Hub (UCM) Data Enrichment Web Services

Oracle Customer Hub (UCM) enables integration with third-party data enrichment providers, such as Acxiom. Oracle Customer Hub (UCM) uses Oracle Application Integration Architecture to provide a prebuilt set of enrichment APIs to enrich consumer records in Oracle Customer Hub (UCM). This service also enriches household and address information for consumer records. The service infrastructure can be extended by small configuration changes to support other third-party customer data enrichment vendors.

Process of Configuring Data Enrichment

To configure for data enrichment, perform the following tasks:

- *Creating the Proxy Business Service*
- *Modifying the Newly Created Proxy Business Service*

- *Modifying Data Enrichment Business Service User Properties*
- *Registering the Data Enrichment Web Services*
- *Configuring the Run-Time Event*

Creating the Proxy Business Service

This task is a step in *Process of Configuring Data Enrichment*.

You perform this task only if you are creating proxy business services using WSDL files provided by a third-party data enrichment Web service.

> **Note:** This configuration task is not required when integrating Oracle Application Integration Architecture with Oracle Customer Hub (UCM). A middleware installation must be completed and set up to handle outbound, data enrichment calls. If you are setting up an integration of Oracle Application Integration Architecture with Oracle Customer Hub (UCM), make sure Oracle Application Integration Architecture is set up and functioning. If Oracle Customer Hub (UCM) is integrated with other middleware, make sure the middleware installation has been completed before proceeding with the following task.

Use the following task to create the proxy business service.

To create the proxy business service

1. From the Siebel Tools File menu, select New Object.
2. In the New Object wizard, click the EAI tab, and click the Web Service Icon.
3. Lock the project Web Service Integration, and lock the Project name.
4. Browse to the location of the WSDL file provided by the third-party data enrichment Web service.

   The run-time data file and log file entries are automatically created.

   > **CAUTION:** Do not check the Process Fault Schema check box, it will produce unintended results.

5. Click Next.
6. Select the Deploy Integration Objects and Proxy Business Services check box.
7. Click Finish.
8. Search the log files for the entry, *Outbound proxy business service definitions to be generated. At the end of the definition, you see the name of the proxy business service that you created.*

   For example:

   ```
   Outbound proxy business service definitions to be generated: ProcessPersonUCMReqABCSImpl
   ```

   In this example, ProcessPersonUCMReqABCSImpl, is the name of the proxy service that was created.
9. When the proxy business service is created, click the Business Services object in the Object Explorer.

Modifying the Newly Created Proxy Business Service

This task is a step in *Process of Configuring Data Enrichment*.

Note: Perform this step only if you are creating proxy business services, using WSDL files provided by a third-party data enrichment Web service.

Use the following task to modify the newly created business service.

To modify the newly created business service

1. In Siebel Tools Business Services screen, query for the newly created proxy business service.
2. In the Object Explorer, navigate to Business Service Method Arguments.
3. Search for arguments of the type, Integration Object.
4. Note the integration object names, then using the Object Explorer, find and delete the objects.

   The default integration objects will not be used. Instead, the newly imported integration objects will be used.
5. Replace the existing integration object name in the Business Service Method Arguments of the proxy service with the new Integration Object name displayed in the following table.

Existing Integration Object Name	New Integration Object Name
Person	SwiPersonIO

Modifying Data Enrichment Business Service User Properties

This task is a step in *Process of Configuring Data Enrichment*.

Note: This configuration task is not required when integrating Oracle Application Integration Architecture with Oracle Customer Hub (UCM).

Use the following task to modify the Data Enrichment business service.

To modify the Data Enrichment business service

1. In Siebel Tools Business Services screen, query for the Data Enrichment business service.
2. In the Object Explorer, navigate to Business Service User Properties.
3. Make the following changes to the user property definitions.

Field	New Value
Proxy Business Service Method	ProcessPerson
Proxy Business Service Name	ProcessPersonUCMReqABCSImpl. This is the name of the Proxy Business Service created previously this process.
RequestTypeContact	ProcessPersonReqMsg:ListOfSwiPersonIO

Field	New Value
	This is the name of the input integration object as mentioned in the method arguments of the proxy business service.
ResponseTypeContact	ProcessPersonRespMsg:ListOfSwiPersonIO This is the name of the output integration object as mentioned in the method arguments of the proxy business service.
UCMIONameContact	SwiPersonIO

Registering the Data Enrichment Web Services

This task is a step in *Process of Configuring Data Enrichment*. Use the following task to register data enrichment Web services.

To register data enrichment Web services

1. If you have an integration of Oracle Application Integration Architecture with Axciom, go to Step 2. If you do not have an integration, go to Step 4.
2. From the Outbound Web Services view, query for the following Oracle Application Integration Architecture Web service: ProcessPersonUCMReqABCSImpl.
3. Change the address of the ports corresponding to this Web service to point to the appropriate Oracle Application Integration Architecture end point.
4. If you do not have Axciom Integration, from the Outbound Web Services view, query for the Web service that you created in the following task, *Creating the Proxy Business Service*.
5. Change the address of the ports corresponding to the Web services to point to the appropriate middleware end point.

Configuring the Run-Time Event

This task is a step in *Process of Configuring Data Enrichment*. Use the following task to configure the run-time event.

To configure the run-time event

1. In the Siebel Mobile Web client, navigate to the Administration - Runtime Events screen, then the Action Sets view.
2. In the Action Sets screen, click New.

3. Enter the following values in the following table.

Field	Value
Name	Choose a name.
Action Type	BusService.
Active	Y.
Sequence	1.
Business Service Name	DataEnrichment.
Business Service Method	Data Enrichment
Conditional Expression	SystemPreference("Enable Data Enrichment") LIKE 'TRUE' AND (GetProfileAttr("Disable Data Enrichment") IS NULL OR GetProfileAttr("Disable Data Enrichment") LIKE 'FALSE')

4. From the Action Sets menu, choose Save Record.
5. Navigate to the Events view, and click New.
6. Create a new event by entering the following values:

Field	Value
Sequence	Use the lowest sequence number. **Note:** Each Object and Event can have several associated events, so provide a unique number for each.
Object Type	BusComp
Object Name	Contact
Event	PreWriteRecord
Action Set Name	The name you provided in Step 3.

7. From the Events menu, select Save Record.
8. Navigate to Administration-Application, then System Preferences.

9. Query for, Enable Data Enrichment, and make sure the System Preference Value is set to True.

Third-Party Data Enrichment Fields

This topic provides information on all the third-party data enrichment fields.

The following table describes the third-party data enrichment fields for the Contact business component.

Fields	Description
Consumer Link	Unique key used for Contact in third-party enrichment vendors.
First Name	First Name of Person.
Middle Name	Middle Name of Person.
Last Name	Last Name of Person.
Name Suffix	Name Suffix of Person.
JOB TITLE	Job Title of Person.
GENDER	Gender of Person.

The following table describes the third-party data enrichment fields for the Personal Address business component.

Fields	Description
Address Link	Unique key used for the addresses in third-party enrichment vendors.
Delivery Point Barcode	Postal delivery point barcode.
Carrier Route Code	A numeric and alpha code that enables mailers to sort mailing in carrier casing sequence order. The eLOT code sorts carrier route coded address records based on the carrier's line of travel. Address lists must be sorted using the eLOT code to receive any carrier route level discounts, except for walk sequence discounts.
Postal Address Flags	Internal footnote flags set by the AddressAbility. (AA)/DPV process. Footnotes are displayed as character codes in the form of Yes or No. These flags appear in the field positions listed as follows: • 01. AA match at ZIP Code level. • 02. AA match at Carrier Route Code level.

Fields	Description
	• 03. AA match at ZIP plus 4 level; ZIP plus 4 applied. • 04. AA match at ZIP plus 4 level; ZIP plus 4 not applied (Undeliverable). • 05. LOT default match. • 06. DPV record match. • 07-08. Not used. • 09. No match; no potentials. • 10. No match; primary number missing. • 11. No match; out of range. • 12. No match; multimatch. • 13. No match, due to match in early-warning system. • 14. Internal use only. • 15. LACSLink match; DPV no match. • 16. Not used. • 17. AA match; DPV no match; ZIP plus 4 dropped. • 18-24. Not used. • 25. ZIP Code match, verified at street number level. • 26. ZIP Code match, verified at street level only. • 27. ZIP Code match verified at town level. • 28. ZIP plus 4 match made for the address only. • 29. ZIP plus 4 match made for the address and secondary (firm) name. • 30. Spellcheck match made. • 31. ZIP plus 4 default match; missing primary number. • 32. ZIP plus 4 default match; missing secondary number.
Postal Address Flags (cont.)	• 33. Rural Route default match. • 34. Rural Route exact match. • 35. High Rise default match. • 36. High Rise exact match. • 37. Unique ZIP Code match. • 38. ZIP Code move match. • 39. Not used. • 40. DPV commercial mail receiving agency (CMRA) match. • 41. Invalid ZIP Code and city identified. • 42. ZIP Code and city conflict. • 43. ZIP code is invalid. The city is valid. • 44. ZIP Code changed or corrected. • 45. City changed or corrected. • 46. State code changed or corrected. • 47. City and state combination is invalid for ZIP Code.

Fields	Description
	• 48. Not used. • 49. Primary number changed or corrected. • 50. Predirectional changed or corrected. • 51. Street changed or corrected. • 52. Suffix changed or corrected. • 53. Postdirectional changed or corrected. • 54. Unit designator changed or corrected. • 55. Secondary number changed or corrected. • 56. Not used. • 57. State blank or invalid. • 58. SCF (Sectional Center Facility) blank or invalid. • 59. Invalid state or SCF combination. • 60. ZIP Code blank or invalid. • 61. Invalid city/ZIP Code combination. • 62. SuiteLink attempted; matched. Suitelink allows the USPS to append a missing suite number to a company's address. • 63. SuiteLink attempted; no match. • 64. Not used.
DSF Delivery Type	None.
DSF Seasonal Indicator	None.
Large Prison Flag	The Zip Code file of all prisons. So if the record matches the prison suppression file the client might want to suppress records that would enter into the prison because of the high probability of fraud.
Acxiom Opt Out Inaccurate Record Flag	When this flag is checked propriatary data sources are invoked. When individuals indicate a desire to receieve offers a value of Highly Sensitive is assigned. It can also indicate that the individual has threatened litigation if additional offers are received.
NCOA Link New Move Effective Date	NCOALink Move Effective Date (YYYYMM).
	NCOALink Move Effective Date (YYYYMM).

The following table describes the third-party data enrichment fields for the UCM Contact Privacy business component.

Fields	Description
Overall Suppression Flag	This field can be used as a single look-up rather than having to look up all fields. It might apply to this business component.
DMA Mail Flag	Direct Marketing Association's email source file for the privacy option preference in email communications.

Fields	Description
DMA Phone Flag	Direct Marketing Association's email source file to the privacy option preference in phone communications
SAG Phone Flag	Refers to all State Attorneys General (SAG). Includes all Do Not Call files from each state combined.
DMA Email Flag	Direct Marketing Association's email file. Its email suppression file. Clients must pass prospecting records to this file prior to running email campaigns.
Acxiom Opt Out Highly Sensitive Flag	The Acxiom Opt-Out Highly Sensitive File is an Acxiom Suppression file that contains names and addresses that are considered highly sensitive. This data includes minors, people who have had their identity stolen, and irate consumers.
Acxiom Opt Out Deceased Flag	Internal Acxiom Consumer-at-Business Opt-out File. The client chooses not to send email to people who are deceased.
Deceased Data Flag	Different data source of the deceased file. Clients want to stop sending marketing information to individuals who are deceased.
Acxiom Opt Out Cons At Bus Flag	The client might want to suppress consumer records at an address. For example, customers provide their marketing address as their home address. However, they requested the product to be shipped to their business address. This field allows customers to suppress the consumer at a business address. For example, you can remove duplicate offers or send information to a preferred address.
Derogatory Data Flag	The record contains negative comments from the customer. Therefore, the client will be cautious in setting spending limits, extending offers, and so on.
Bankruptcy Data Flag	The customer is involved in bankruptcy proceedings. Therefore, the client will be cautious in setting spending limits, extending offers, and so on.
Tax Lien Judgment Flag	Tax Lien and Judgment File. Indicates that the customer has a tax lien on his or her property. Therefore, the client will not want to extend additional offers or credit to the customer.
FTC Do Not Call Flag	Different data sources for the Do Not Call suppression. Prospect phone numbers must be matched against this file prior to performing telemarketing exercises.
Wireless Flag	Direct Marketing Association's wireless phone preference file. Legislation forbids telemarketing to cell phones. This flag must be set prior to telemarketing activities.

The following table describes the third-party data enrichment fields for the Household business component.

Fields	Description
Household Link	Unique key used for a household in third-party enrichment vendors.
Home Owner/Renter	Indicates if a household is owner occupied or renter occupied.

Fields	Description
Length Of Residence	Indicates the total time a household has lived at the current address. Specified in ranges.
1st Individual Age	Indicates the age of the first individual listed in the household.
Adults in Household	Indicates the number of adults in the household. An adult is anyone 18 years old or older living in a household.
1st Individual Education	Indicates the education of the first individual in listed in the household.
Personicx LifeStage Groups	*Personicx* is a household-level segmentation method that places each U.S. household into one of 70 segments. The segments are based on specific consumer and demographic characteristics.

Oracle Data Governance Manager Web Services

The following table describes the SWIContactServices operations.

Name	Description
SWIContactServicesInsertOrUpdate	Calls Insert or Update method used for inserting a new contact along with the child details into the Siebel database. If the contact is already present in the Siebel database, the Contact will be updated with the details from the request message.
SWIContactServicesQueryByExample	Calls the Query by Example method to Query Contact details along with the child details from the Siebel database.
SWIContactServicesSynchronize	Calls the Synchronize method to synchronize the contact in the request message with the contact details in Siebel. If the child details that are part of the request message are not present in the Siebel database, they will be Inserted and if any additional child are present in the database and not part of request message, those will be deleted from the database.

The following table describes the SWIContactServicesInsertOrUpdate request message.

Node	Description	Type
Alias	Alias of the contact record.	String
CellularPhone	Mobile phone number of the contact.	String
ContactPersonTitle	Contact's title.	String

Node	Description	Type
CurrencyCode	Contact's account currency code.	String
DateofBirth	Contact's birth date.	String
EmailAddress	Contact's email address.	String
FaxPhone	Contact's fax number.	String
FirstName	Contact's first name.	String
Gender	Contact's gender.	String
HomePhone	Contact's home phone number.	String
JobTitle	Contact's job title.	String
LastName	Contact's last name.	String
MM	Mr. or Mrs.	String
MaritalStatus	Contact's marital status.	String
MiddleName	Contact's middle name.	String
RowId	Row Id of Siebel Contact record	String
SocialSecurityNumber	Contact's social security number.	String
Status	Contact record's status.	String
WorkPhone	Contact's work phone number.	String
AIAIntegrationId	Id value used for mapping the request message and response message contact details.	String
ListofOrganization	Container for Organization	String
PersonalAddress	Contact's home address.	String
Id	Siebel row Id of the Contact record.	String

Node	Description	Type
AddressName	Contact's address name.	String
City	Contact's city of residence.	String
Country	Contact's country of residence.	String
PostalCode	Contact's postal code.	String
Province	Contact's province, if applicable.	String
State	Contact's state of residence.	String
StreetAddress1	Line 1 details for the contact.	String
StreetAddress2	Line 2 details for the contact.	String
AIAIntegrationId	Id value used for mapping the request and response message.	String

The following table describes the SWIContactServicesInsertOrUpdate response message.

Node	Description	Type
Alias	Alias of contact record.	String
CellularPhone	Contact's mobile phone number.	String
ContactPersonTitle	Contact's title.	String
CurrencyCode	Contact account currency code.	String
DateofBirth	Contact's birth date.	String
EmailAddress	Contact's email address.	String
FaxPhone	Contact's fax number.	String
FirstName	Contact's first name.	String
Gender	Contact's gender.	String

Node	Description	Type
HomePhone	Contact's home phone number.	String
JobTitle	Contact's job title.	String
LastName	Contact's last name.	String
MM	Mr. or Mrs.	String
MaritalStatus	Contact's marital status.	String
MiddleName	Contact's middle name.	String
RowId	Row Id of Siebel Contact record	String
SocialSecurityNumber	Contact's social security number.	String
Status	Contact status.	String
WorkPhone	Contact's work phone number.	String
AIAIntegrationId	Id value used for mapping the request message and response message contact details.	String
ListofOrganization	Container for organization record.	String
PersonalAddress	Contact's home address.	String
Id	Row Id of the Contact record In Siebel.	String
AddressName	Address name	String
City	Contact's city.	String
Country	Contact's country.	String
PostalCode	Contact's postal code.	String
Province	Contact's province (if applicable).	String
State	Contact's state.	String
StreetAddress1	Line 1 details for the contact.	String

Node	Description	Type
StreetAddress2	Line 2 details for the contact.	String
AIAIntegrationId	Id value used for mapping the request and response message.	String

The following table describes the SWIContactServicesQueryByExample request message.

Node	Description	Type
RowId	Row Id of Siebel Contact record	String
ListOfAccount	List of Account records.	String
ListofPersonalAddress	List of personal address records of contact.	String
PersonalAddress	Contact's personal address.	String

The following table describes the SWIContactServicesQueryByExample response message.

Node	Description	Type
Id	Row Id.	String
Alias	Alias of contact record.	String
CellularPhone	Contact's mobile phone number.	String
ContactPersonTitle	Contact's title.	String
CurrencyCode	Contact account currency code.	String
DateofBirth	Contact's birth date.	String
EmailAddress	Contact's email address.	String
FaxPhone	Contact's fax number.	String
FirstName	Contact's first name.	String
Gender	Contact's gender.	String

Node	Description	Type
HomePhone	Contact's home phone number.	String
JobTitle	Contact's job title.	String
LastName	Contact's last name.	String
MM	Mr. or Mrs.	String
MaritalStatus	Contact's marital status.	String
MiddleName	Contact's middle name.	String
MotherMaidenName	Contact's mother's maiden name.	String
PrimaryOrganizationId	Primary organization for contact.	String
RowId	Row Id in Siebel data table.	String
SocialSecurityNumber	Contact's social security number.	String
Status	Status of contact record.	String
WorkPhone	Contact's work telephone number.	String
ListofAccount	Container for Account.	Integration Component
ListofComInvoiceProfile	Container for invoice profile.	Integration Component
ListofOrganization	Container for Organization.	Integration Component
ListofPersonalAddress	Container for personal address records.	Integration Component
PersonalAddress	Contact's personal address.	String
ListofUCMContactPrivacy	Privacy details for Contact record.	Integration Component

The following table describes the SWIContactServicesSynchronize request message.

Node	Description	Type
Id	Row Id.	String
Alias	Alias of contact record.	String
CellularPhone	Contact's mobile phone number.	String
ContactPersonTitle	Contact's title.	String
CurrencyCode	Contact's account currency code.	String
DateofBirth	Contact's birth date.	String
EmailAddress	Contact's email address.	String
FaxPhone	Contact's fax number.	String
FirstName	Contact's first name.	String
Gender	Contact's gender.	String
HomePhone	Contact's home phone number.	String
JobTitle	Contact's job title.	String
LastName	Contact's last name.	String
MM	Mr. or Mrs.	String
MaritalStatus	Contact's marital status.	String
MiddleName	Contact's middle name.	String
MotherMaidenName	Contact's mother's maiden name.	String
PrimaryOrganizationId	Primary organization for contact.	String
RowId	Row Id in Siebel data table.	String
SocialSecurityNumber	Contact's social security number.	String
Status	Status of contact record.	String

Node	Description	Type
WorkPhone	Contact's work telephone number.	String
AIAIntegrationId	Id value used for mapping the request message and response message contact details.	String
ListofOrganization	Container for Organization.	Integration Component
ListofPersonalAddress	Container for personal address records.	Integration Component

The following table describes the SWIContactServicesSynchronize response message.

Node	Description	Type
Id	Row Id.	String
Alias	Alias of contact record.	String
CellularPhone	Contact's mobile phone number.	String
ContactPersonTitle	Contact's title.	String
CurrencyCode	Contact's account currency code.	String
DateofBirth	Contact's birth date.	String
EmailAddress	Contact's email address.	String
FaxPhone	Contact's fax number.	String
FirstName	Contact's first name.	String
Gender	Contact's gender.	String
HomePhone	Contact's home phone number.	String
JobTitle	Contact's job title.	String
LastName	Contact's last name.	String
MM	Mr. or Mrs.	String

Node	Description	Type
MaritalStatus	Contact's marital status.	String
MiddleName	Contact's middle name.	String
MotherMaidenName	Contact's mother's maiden name.	String
PrimaryOrganizationId	Primary organization for contact.	String
RowId	Row Id in Siebel data table.	String
SocialSecurityNumber	Contact's social security number.	String
Status	Status of contact record.	String
WorkPhone	Contact's work telephone number.	String
AIAIntegrationId	Id value used for mapping the request message and response message contact details.	String
ListofOrganization	Container for Organization.	Integration Component
ListofPersonalAddress	Container for personal address records.	Integration Component

The following table describes the SWIContactService service object.

Siebel Repository Name	Boundary Object Type	Class
SWI Contact Service	Business Service	CSSEAIDataSyncService

The following table describes the SWIContactService data object.

Siebel Repository Name	External Name
SWIContactIO	SWI Contact

The following table describes the SWICustomerPartyService operations.

Name	Description
SWICustomerPartyInsertOrUpdate	Calls the Insert or Update method used for inserting a new account along with the child details into the Siebel database. If the account is already present in the Siebel database the account will be updated with any new details contained in the request message.
SWICustomerPartyQueryByExample	Calls the Query by Example method to query the Siebel database for account details along with any child details.
SWICustomerPartySynchronize	Calls Synchronize method to synchronize the account record contained in the request message with the account details in Siebel database. If the child details that are part of request message are not present in the Siebel database, the details will inserted and if any additional child are present in the Siebel database, are not part of request message those details will be deleted from the database.

The following table describes the SWICustomerPartyInsertOrUpdate request message.

Note: All nodes in the table bar the first node apply to SwicustomerpartyIO.

Node	Description	Type
ListofSwicustomerpartyIO	An instance of the Integration Object ListofSwicustomerpartyIO.	Integration Object
Account	Account name.	String
AccountStatus	Status of account record.	String
DUNSNumber	Dun and Bradstreet data universal numbering system number.	String
MainFaxNumber	Main fax number of account.	String
MainPhoneNumber	Main phone number of account.	String
Name	Account name.	String
HomePage	Account's home page address.	String
AIAIntegrationId	Id value used for mapping the request message and response message account details.	String
Location	Account's location.	String
RowId	Row Id of Siebel Account record.	String

Node	Description	Type
ListofCutAddress	Optional. Business address.	Integration Component
ListofContact	Contact details	Integration Component
ListofInternalDivision	Internal division details.	Integraton Component
City	City where account is located.	String
Country	Country where acccount is located.	String
County	County where acccount is located.	String
PostalCode	Postal code of account.	String
Province	Province (if applicable) where account is located.	String
AIAIntegrationId	Contact's social security number.	String
Status	Contact status.	String
WorkPhone	Contact's work phone number.	String
AIAIntegrationId	Id value used for mapping the request message and response message contact details.	String
ListofOrganization	Container for organization record.	String
PersonalAddress	Contact's home address.	String
Id	Row Id of the Contact record in Siebel data table.	String
AddressName	Address name	String
City	Contact's city.	String
Country	Contact's country.	String
PostalCode	Contact's postal code.	String
Province	Contact's province (if applicable).	String

Node	Description	Type
State	Contact's state.	String
StreetAddress1	Line 1 details for the contact.	String
StreetAddress2	Line 2 details for the contact.	String
AIAIntegrationId	Id value used for mapping the request and response message.	String

This table describes the SWICustomerPartyInsertOrUpdate response message.

Note: All nodes in the table bar the first node apply to SwicustomerpartyIO.

Node	Description	Type
ListofSwicustomerpartyIO	An instance of the Integration Object ListofSwicustomerpartyIO.	Integration Object
Account	Account name.	String
AccountStatus	Status of account record.	String
DUNSNumber	Dun and Bradstreet data universal numbering system number.	String
HomePage	Account's home page address.	String
Location	Account's location.	String
MainFaxNumber	Account's main fax number.	String
MainPhoneNumber	Account's main phone number.	String
Name	Account name.	String
AIAIntegrationId	Id value used for mapping the request message and response message account details.	String
ListofCutAddress	Optional. Business address.	Integration Component
ListofContact	Contact details	Integration Component

Node	Description	Type
ListOfInternalDivision	Internal division details.	Integration Component

The following table describes the SWICustomerPartyQueryByExample request message.

Note: All nodes in the table bar the first node apply to SwicustomerpartyIO.

Node	Description	Type
ListofSwicustomerpartyIO	An instance of the Integration Object ListofSwicustomerpartyIO.	Integration Object
RowId	Row Id of account in Siebel data table.	String
ListofCutAddress	Optional. Business address.	Integration Component

The following table describes the SWIContactServicesQueryByExample response message.

Note: All nodes in the table bar the first node apply to SwicustomerpartyIO.

Node	Description	Type
ListofSwicustomerpartyIO	An instance of the Integration Object ListofSwicustomerpartyIO.	Integration Object
Account	Account name.	String
AccountStatus	Status of account record.	String
AccountTypeCode	Dun and Bradstreet data universal numbering system number.	String
CurrencyCode	Account's currency code.	String
DUNSNumber	Dun and Bradstreet data universal numbering system number.	String
HomePage	Homepage of account	String
IntegrationId	Integration Id.	String
Location	Account location.	String

Node	Description	Type
MainFaxNumber	Main fax number of account.	String
MainPhoneNumber	Main phone number of account.	String
Name	Account name.	String
ParentAccountId	Account Id of parent account.	String
PrimaryAddressId	Primary address of account.	String
PrimaryBillToAddressId	Primary billing address of account.	String
PrimaryContactId	Primary Contact Id for the account.	String
PrimaryOrganizationId	Primary Organization Id for the account.	String
PrimaryShipToAddress	Primary shipping address for the account.	String
RowId	Row Id in Siebel data table.	String
Type	Account type.	String
ListofCutAddress	Optional. Business address.	Integration Component
ListofContact	Contact details	Integration Component
ListofInternalDivision	Internal division details.	Integraton Component

The following table describes the SWICustomerPartySynchronize request message.

Note: All nodes in the table bar the first node apply to SwicustomerpartyIO.

Node	Description	Type
ListofSwicustomerpartyIO	An instance of the Integration Object ListofSwicustomerpartyIO.	Integration Object
Account	Account name.	String
AccountStatus	Status of account record.	String

Node	Description	Type
DUNSNumber	Dun and Bradstreet data universal numbering system number.	String
MainFaxNumber	Main fax number of account.	String
MainPhoneNumber	Main phone number of account.	String
Name	Account name.	String
HomePage	Account's home page address.	String
AIAIntegrationId	Id value used for mapping the request message and response message account details.	String
Location	Account's location.	String
RowId	Row Id of Siebel Account record.	String
ListofCutAddress	Optional. Business address.	Integration Component
ListofContact	Contact details	Integration Component
ListofInternalDivision	Internal division details.	Integraton Component
City	City where account is located.	String
Country	Country where acccount is located.	String
County	County where acccount is located.	String
PostalCode	Postal code of account.	String
Province	Province (if applicable) where account is located.	String
AIAIntegrationId	Contact's social security number.	String
Status	Contact status.	String
WorkPhone	Contact's work phone number.	String
AIAIntegrationId	Id value used for mapping the request message and response message contact details.	String

Node	Description	Type
ListofOrganization	Container for organization record.	String
PersonalAddress	Contact's home address.	String
Id	Row Id of the Contact record in Siebel data table.	String
AddressName	Address name	String
City	Contact's city.	String
Country	Contact's country.	String
PostalCode	Contact's postal code.	String
Province	Contact's province (if applicable).	String
State	Contact's state.	String
StreetAddress1	Line 1 details for the contact.	String
StreetAddress2	Line 2 details for the contact.	String
AIAIntegrationId	Id value used for mapping the request and response message.	String

The following table describes the SWICustomerPartySynchronize response message.

Note: All nodes in the table bar the first node apply to SwicustomerpartyIO.

Node	Description	Type
ListofSwicustomerpartyIO	An instance of the Integration Object ListofSwicustomerpartyIO.	Integration Object
Account	Account name.	String
AccountStatus	Status of account record.	String
DUNSNumber	Dun and Bradstreet data universal numbering system number.	String

Node	Description	Type
HomePage	Account's home page address.	String
Location	Account's location.	String
MainFaxNumber	Account's main fax number.	String
MainPhoneNumber	Account's main phone number.	String
Name	Account name.	String
AIAIntegrationId	Id value used for mapping the request message and response message account details.	String
ListofCutAddress	Optional. Business address.	Integration Component
ListofContact	Contact details	Integration Component
ListOfInternalDivision	Internal division details.	Integraton Component

The following table describes the Service Object.

Siebel Repository Name	Boundary Object Type	Class
SWI Customer Party Service	Business Service	CSSEAIDataSyncService

The following table describes the Data Object.

Siebel Repository Name	External Name
SWICustomerPartyIO	Account

The following table describes the SWIMergeServiceMerge operation.

Name	Description
SWIMergeServicesMerge	Calls Merge method of the Business Service SWI Merge Service which is used for merging one or more accounts to another account record. This Web service merges child records of the losing objects to the wining object and deletes the losing object records. Takes the Business Object name, Business Component name, one wining record's Id value and one or more losing record's Id values as the arguments in .xml format.

The following table describes the SWIMergeServicesMerge request message.

Note: All nodes in the table bar the first node apply to ListOfSwimergeobjectIO.

Node	Description	Type
ListOfSwimergeobjectIO	An instance of the Integration Object ListofSwimergeobjectIO.	Integration Object
BusinessComponentName	Name of business component.	String
BusinessObjectName	Name of business object.	String
MergeId	Id number assigned to merged item.	String
ListofSwiLosingObject	Container element for the Losing Object details which is an instance of the VBC SWI Losing Object. Contains the details of the object that is merged to another object.	Integration Component
ListOfSwiWinningObject	Container element for the Winning Object details which is an instance of the VBC SWI Winning Object. Contains the details of the object that is merged to another object.	Integration Component

The following table describes the SWIMergeServicesMerge response message.

Note: All nodes in the table bar the first node apply to ListOfSwimergeobjectIO.

Node	Description	Type
ListOfSwimergeobjectIO	An instance of the Integration Object ListofSwimergeobjectIO.	Integration Object
BusinessComponentName	Name of business component.	String
BusinessObjectName	Name of business object.	String
MergeId	Id number assigned to merged item.	String
ListofSwiLosingObject	Container element for the Losing Object details which is an instance of the VBC SWI Losing Object. Contains the details of the object that is merged to another object.	Integration Component

Node	Description	Type
ListOfSwiWinningObject	Container element for the Winning Object details which is an instance of the VBC SWI Winning Object. Contains the details of the object that is merged to another object.	Integration Component

The following table describes the SWIMergeService service object.

Siebel Repository Name	Boundary Object Type	Class
SWIMergeService	Business Service	CSSMergeService

The following table describes the SWIMergeService data object.

Siebel Repository Name	External Name
SWIMergeObjectIO	SWI Merge Object

The following table describes the SWIProductImport operations.

Name	Description
SWIProductImportUpsert	Calls 'Insert or Update' method of the Business Service 'SWI Product Import' used for inserting a new product details in to Siebel. If the product is already present in Siebel the product will be updated with the details from the request message.

The following table describes the SWIProductImportUpsert request message.

Node	Description	Type
ActiveFlag	Indicates whether record is active.	String
BillableFlag	Indicates whether record is billable.	String
BusinessUnitId	Id of business unit associated with product record.	String
Description	Description.	String
EffectiveEndDate	Effective end date of record.	String
EffectiveStartDate	Effective start date of record.	String

Node	Description	Type
ExternalInventorySystemRef	Reference number from external inventory system.	String
IntegrationId	Integration Id.	String
InventoryFlag	Flag to indicate if product is tracked in inventory.	String
LeadTime	Lead time for product delivery.	String
ProductCost	Cost of product.	String
ProductName	Name of product.	String
ProductSerializedFlag	Serialized flag. Indicates if instances of this product are tracked as serialized assets or simply as quantities of product. Instances of a serialized product will each carry a unique serial number (or license number, policy number, and so on). Quantities of non-serialized assets are sometimes tracked as lots which may have a unique lot number.	String
ProductVersion	Version of product.	String
ReleaseFlag	Flag which indicates whether a product will be released at the time of a synchronization.	String
SalesProductFlag	Sales product flag. Sales products appear on the product picklist when a user defines products for an opportunity.	String
ServiceTerms	Service terms associated with product.	String
ShipFlag	Flag to indicate whether a product has to be shipped.	String
Status	Status of product. Active or Inactive.	String
StructureType	Depending on the product type, users may be allowed to create a simple bill of materials without using Siebel Product Configurator.	String
UnitofMeasure	Unit of measure.	String
ListofProductInvloc	Location of product in inventory.	String

The following table describes the SWIProductImportUpsert response message.

Node	Description	Type
Id	Product Id.	String
ActiveFlag	Indicates whether record is active.	String
AutoExplodeFlag	Indicates whether or not the product should be automatically ungrouped and enforce ungrouped quantity equal to 1.	String
BillableFlag	Indicates whether record is billable.	String
BusinessUnitId	Id of business unit associated with product record.	String
CheckEligibilityFlag	Indicates whether or not to apply eligibility and compatibility rules for this product.	String
Description	Description of product.	String
EffectiveEndTime	Effective end date of record.	String
EffectiveStartTime	Effective start date of record.	String
ExternalInventorySystemRef	Reference number from external inventory system.	String
InclusiveEligibilityFlag	Indicates whether or not to process a particular product for inclusive eligibility rules. Inclusive eligibility rules are defined in the eligibility matrix with matrix rule type code set as Inclusive.	String
IntegrationId	Integration Id.	String
InventoryFlag	Flag which indicates if a product is tracked in inventory.	String
LeadTime	Lead time for product delivery.	String
OrderableFlag	Flag which indicates whether a product is orderable.	String
PriceType	Price type, such as One-time, Recurring, Usage and so on.	String
PrimaryOrganizationId	Id of primary organization associated with product record.	String
ProductCost	Cost of product.	String
ProductId	Id of product.	String
ProductName	Name of product.	String

Node	Description	Type
ProductSerializedFlag	Serialized flag. Indicates if instances of this product are tracked as serialized assets or simply as quantities of product. Instances of a serialized product will each carry a unique serial number (or license number, policy number, and so on). Quantities of non-serialized assets are sometimes tracked as lots which may have a unique lot number.	String
ProductType	Product type.	String
ProductVersion	Product version.	String
ReleaseFlag	Flag which indicates whether a product will be released at the time of a synchronization.	String
SalesProductFlag	Sales product flag. Sales products appear on the product picklist when a user defines products for an opportunity.	String
ServiceInstanceFlag	Flag to indicate whether a product requires a metering point.	String
ServiceProductFlag	Sales and Service flag. Sales service products are services that are priced based on the tangible product for which they are purchased (an extended warranty, for example). These products appear on the picklist that pops up when the user clicks the Service button for a tangible product on a quote. When you choose a sales service product from the list, a new quote item for the product is added to the quote and priced according to the value of the tangible product to which it applies.	String
ShipFlag	Flag to indicate whether a product needs to be shipped.	String
Status	Status of product order.	String
StructureType	Depending on the product type, users may be allowed to create a simple bill of materials without using Siebel Product Configurator.	String
TrackAsAssetFlag	Flag to indicate that a product should be tracked as an asset.	String
Type	Product type.	String
UnitofMeasure	Unit of measure.	String
VendorId	Id of product vendor.	String
ListofProductInvLoc	Location of product in inventory.	String

The following table describes the SWIProductImport service object.

Siebel Repository Name	Boundary Object Type	Class
SWI Product Import	Business Service	CSSCMUProdWebSvc

The following table describes the SWIProductImport data object.

Siebel Repository Name	External Name
SWIProductIntegrationIO	SWI Product Integration

Services

The following Web services are used to perform operations for Oracle Data Governance Manager. These Web services include:

- *SystemsRegistration Service*
- *MasterService*
- *ConsolidateService*
- *ShareService*
- *CleanseCompletenessService*
- *GovernService*

SystemsRegistrationService

Use this Web service to query details of all systems registered with Oracle Customer Hub (UCM).

SystemsRegistrationService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
getSystemsRegistrationInfo	Queries the details of all systems registered with Oracle Customer Hub (UCM).

Request Message Description: getSystemsRegistrationInfo

There is no request message for this operation.

Response Message Description: getSystemsRegistrationInfo

For a description of this response message, see the following table.

Node	Description	Type
SwiSystemInfoIO	An instance of the integration object, SwiSystemInfoIO.	Integration Object
errorMessage	Specific error message.	String

SystemsRegistrationService Application Interface

This topic describes the application objects called by this web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
UCM Data Governance Manager	Business Service	CSSUCMDataGovernanceManagerService

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
getSystemsRegistrationInfo	getSystemsRegistrationInfo

ConsolidateListImportService

The List Import function serves as a standalone Web service. It is invoked from Oracle Data Governance Manager. It creates an import job and activates it. The ConsolidateListImportService can be used with any supported application within your system to load data into Oracle Customer Hub (UCM) by calling the Web service and using the required inputs.

ConsolidateListImportService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
obtainTemplateList	Returns details from all Import Mappings that are currently present in the List Import function. These mappings are then used to import records from text or XML files into Oracle Customer Hub (UCM).
submitNewImport	Takes information about a List Import job, such as import mapping, file stream, data type, and so on. It then creates and submits the import job.

Request Message Description: obtainTemplateList

There is no request message for this Web service.

Response Message Description: obtainTemplateList

For a description of this response message, see the following table.

Node	Description	Type
SwiImportMapsIO	An instance of the integration object, SwiImportMaps.	Integration Object
errorMessage	Specific error message.	String

Request Message Description: submitNewImport

For a description of this request message, see the following table.

Node	Description	Type
ImportJobsParams (UCMImportJob)	Specifies the parameters for the UCM import job.	String.

Response Message Description: obtainTemplateList

There is no response message for this Web service.

ConsolidateListImportService Service Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
UCMListImportService	Business Service	CSSUCMListImportService

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
UCMImportJobIO	UCM Import Job

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
obtainTemplateList	obtainTemplateList
submitNewImport	submitNewImport

MasterService

This Web service is used with the Master module to query and return mastered records from a given period.

MasterService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
getMasterEntitiesCount	Returns the number of mastered Account or Contact records in a given period. The input is given in the form of an integration object where the object and search specification are specified.
getMasterOrganizationDetails	Returns details of the mastered Account records from a given period.

Name	Description
getMasterPersonDetails	Returns details of the mastered Contact records from a given period.

Request Message Description: getMasterEntitiesCount

For a description of this request message, see the following table.

Node	Description	Type
startDateTime	Start date and time for a given job.	String
endDateTime	End date and time for a given job.	String
SwiRecordCountIO	An instance of the integration object, SwiRecordCount.	Integration Object

Response Message Description: getMasterEntitiesCount

For a description of this response message, see the following table.

Node	Description	Type
SwiRecordCountIOResults	An instance of the integration object, SwiRecordCountResults.	Integration Object
errorMessage	Specific error message.	String

Request Message Description: getMasterOrganizationDetails

For a description of this request message, see the following table.

Node	Description	Type
startDateTime	Start date and time for a given job.	String
endDateTime	End date and time for a given job.	String
StartRowNum	The row number of the total number of records from which the records are to be given.	String
PageSize	Indicates the number of records to be returned.	String
SortSpec	The sort criteria applied to the business component.	String

Node	Description	Type
SearchSpec	Search specification for the Account or Contact business component.	String
NewQuery	Starts a new query.	String

Response Message Description: getMasterOrganizationDetails

For a description of this response message, see the following table.

Node	Description	Type
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
NumOutputObjects	Number of records being returned in this iteration of the request result set.	String
SwiOrgIO	An instance of the integration object, SwiOrgIO.	Integration Object

Request Message Description: getMasterPersonDetails

For a description of this request message, see the following table.

Node	Description	Type
startDateTime	Start date and time for a given job.	String
endDateTime	End date and time for a given job.	String
StartRowNum	The row number of the total number of records from which the records are to be given.	String
NewQuery	Starts a new query.	String
PageSize	Indicates the number of records to be returned.	String
SortSpec	The sort criteria applied to the business component.	String
SearchSpec	Search specification for the Account or Contact business component.	String

Response Message Description: getMasterPersonDetails

For a description of this response message, see the following table.

Node	Description	Type
errorMessage	Specific error message.	String
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
NumOutputObjects	Number of records being returned in this iteration of the request result set.	String

MasterService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
UCM Data Governance Manager	BusinessService	CSSSwiUCMDGMgrService

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
SwiOrgIO	Account
SwiPerIO	Contact

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
getMasterEntitiesCount	getMasterEntitiesCount
getMasterOrganizationDetails	getMasterOrganizationDetails
getMasterPersonDetails	getMasterPersonDetails

ConsolidateService

This Web service is used with the Consolidate module to query and return batch-report information from account and contact batch reports run in a given period.

ConsolidateService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
getBatchProcessReport	Returns the Completed, In Progress, and Pending batches from a given period. Return the details (System, BatchID, Description and so on) of a batch in the form of an Integration Object .
getSpecificBatchReport	Returns the count of Completed, Pending and Rejected records in a particular batch (matching with Batch ID).
getCompletedRecordsBatchReport	Returns the number of Inserts or Updates that occurred as part of a completed batch job. The records are mastered as part of the batch job, and a field called Actual Operation stores the information of the operation that took place.
getOrgRejectRecordsDetailReport	Returns the details of all Account records that were rejected as part of a particular batch.
getPerRejectRecordsDetailReport	Returns the details of all Contact records that were rejected as part of a particular batch.
updateOrgBatchRejectedRecords	Allows the user to submit corrected information to the field values of the rejected account records. Then the values in the SDH table are updated. This operation also sets the UCM Type Code to Queued or Batch so that these corrected records can be picked up in the next Batch process iteration.
updatePerBatchRejectedRecords	Allows the user to submit the corrected information to the field values of the rejected contact records. Then the values in the SDH table are updated. This operation also sets the UCM Type Code to Queued or Batch, so that these corrected records can be picked up in the next batch process iteration.

Request Message Description: getBatchProcessReport

For a description of this request message, see the following table.

Node	Description	Type
startDateTime	Start date and time for a given job.	String
endDateTime	End date and time for a given job.	String

Response Message Description: getBatchProcessReport

For a description of this response message, see the following table.

Node	Description	Type
SwiBatchInfoIO	An instance of the integration object, SwiBatchInfo.	Integration Object
errorMessage	Specific error message.	String

Request Message Description: getSpecificBatchReport

For a description of this request message, see the following table.

Node	Description	Type
BatchID	Required. Batch ID number of the batch report.	String
Object	Account or Contact.	String

Response Message Description: getSpecificBatchReport

For a description of this response message, see the following table.

Node	Description	Type
CompletedRecordsCount	Number of records successfully completed in a batch report.	String
PendingRecordsCount	Number of records pending for a batch report.	String
RejectedRecordsCount	Number of records rejected during a batch report.	String
errorMessage	Specific error message.	String

Node	Description	Type

Request Message Description: getCompletedRecordsBatchReport

For a description of this request message, see the following table.

Node	Description	Type
BatchID	Required. Batch ID number of the batch report.	String
Object	Account or Contact.	String

Response Message Description: getCompletedRecordsBatchReport

For a description of this response message, see the following table.

Node	Description	Type
NumofInserts	Number of completed inserts in the batch report.	String
NumofUpserts	Number of completed upserts in the batch report.	String
errorMessage	Specific error message.	String

Request Message Description: getOrgRejectRecordsDetailReport

For a description of this request message, see the following table.

Node	Description	Type
BatchID	Required. Batch ID number of the batch report.	String
StartRowNum	The row number of the total number of records from which the records are to be given.	String
PageSize	Indicates the number of records to be returned.	String
SortSpect	The sort criteria applied to the business component.	String
NewQuery	Starts a new query.	String

Response Message Description: getOrgRejectRecordsDetailReport

For a description of this response message, see the following table.

Node	Description	Type
SwiBatchOrgIO	An instance of the integration object, SwiBatchOrg.	Integration Object
errorMessage	Specific error message.	String
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
NumOutputObjects	Number of records being returned in this iteration of the request result set.	String

Request Message Description: getPerRejectRecordsDetailReport

For a description of this request message, see the following table.

Node	Description	Type
BatchID	Required. Batch ID number of the batch report.	String
StartRowNum	The row number of the total number of records from which the records are to be given.	String
PageSize	Indicates the number of records to be returned.	String
SortSpect	The sort criteria applied to the business component.	String
NewQuery	Starts a new query.	String

Response Message Description: getPerRejectRecordsDetailReport

For a description of this response message, see the following table.

Node	Description	Type
SwiBatchPerIO	An instance of the integration object, SwiBatchPer.	Integration Object
errorMessage	Specific error message	String
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String

Node	Description	Type
NumOutputObjects	Number of records being returned in this iteration of the request result set.	String

Request Message Description: updateOrgBatchRejectedRecords

For a description of this request message, see the following table.

Node	Description	Type
BatchID	Required. Batch ID number of the batch report.	String
SwiBatchOrgIO	An instance of the integration object, SwiBatchOrg.	Integration Object

Response Message Description: updateOrgBatchRejectedRecords

For a description of this response message, see the following table.

Node	Description	Type
errorMessage	Specific error message	String

Request Message Description: updatePerBatchRejectedRecords

For a description of this request message, see the following table.

Node	Description	Type
BatchID	Required. Batch ID number of the batch report.	String
SwiBatchPerIO	An instance of the integration object, SwiBatchPer.	Integration Object

Response Message Description: updatePerBatchRejectedRecords

For a description of this response message, see the following table.

Node	Description	Type
errorMessage	Specific error message.	String

ConsolidateService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
UCM Data Governance Manager	Business Service	CSSSwiUCMDGMgrService

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
SwiBatchInfoIO	SwiBatchInfoIO
SwiBatchOrgIO	UCM Account Source Data and History
SwiBatchPerIO	UCM Contact Source Data and History

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
getBatchProcessReport	getBatchProcessReport
getSpecificBatchReport	getSpecificBatchReport
getCompletedRecordsBatchReport	getCompletedRecordsBatchReport
getOrgRejectRecordsDetailReport	getOrgRejectRecordsDetailReport
getPerRejectRecordsDetailReport	getPerRejectRecordsDetailReport
updateOrgBatchRejectedRecords	updateOrgBatchRejectedRecords

Operation	Method
updatePerBatchRejectedRecords	updatePerBatchRejectedRecords

ShareService

This Web service is used with the Share module to return modified, merged, and unmerged records for Account and Contact business objects run in a given period.

ShareService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
getShareInfo	Returns modified, merged and unmerged records (Account or Contact) in the given period. Only real-time operations are counted.

Request Message Description: getShareInfo

For a description of this request message, see *ShareService Operations*.

Node	Description	Type
startDateTime	Start date and time for a given job.	String
endDateTime	End date and time for a given job.	String
Object	Account or Contact.	String

Response Message Description: getShareInfo

For a description of this response message, see *ShareService Operations*.

Node	Description	Type
MergeRecCount	Total of merged records.	String
UnmergeRecCount	Total of unmerged records.	String.

Node	Description	Type
ModifiedCount	Total number of modified records.	String
errorMessage	Specific error message.	String

ShareService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

ServiceObject (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
UCM Data Governance Manager	Business Service	CSSSwiUCMDGMgrService

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
getShareInfo	getShareInfo

CleanseCompletenessService

This Web service is used to query information on incomplete records of both Oracle Customer Hub (UCM) and source systems. Information is returned as a percentage of the total number of incomplete records.

CleanseCompletenessService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
getHubCompleteComplianceInfo	Queries base table records which, based on the date and the object, are found to be incomplete based on user specified criteria.

Name	Description
getSourceCompleteComplianceInfo	Queries SDH table source records which are found to be incomplete based on user specified criteria. Fields used to make the determination are: date, object and source system.
getSystemsRegistrationInfo	Queries for and returns information from systems registered with Oracle Customer Hub (UCM).
getHubOrgIncompleteDetail	Queries Oracle Customer Hub (UCM) for details of the incomplete records from the account base table. The query is based on user specified criteria.
getHubPerIncompleteDetail	Queries Oracle Customer Hub (UCM) for details of the incomplete records from the contact base table. The query is based on user specified criteria.
getSrcOrgIncompleteDetail	Queries Oracle Customer Hub (UCM) for details of the source incomplete records from the account base table. The query is based on user specified criteria.
getSrcPerIncompleteDetail	Queries Oracle Customer Hub (UCM) for details of the source incomplete records from the contact base table. The query is based on user specified criteria.

Request Message Description: getHubCompleteComplianceInfo

For a description of this request message, see the following table.

Node	Description	Type
startDateTime	Start date and time for a given job.	String
endDateTime	End date and time for a given job.	String
ObjectName	Business object name of the completeness report. It is one of the following: Contact or Account.	String
SwiOrgIO	An instance of the integration object SWIOrg.	Integration Object
SwiPerIO	An instance of the integration object SwiPer.	Integration Object

Response Message Description: getHubCompleteComplianceInfo

For a description of this response message, see the following table.

Node	Description	Type
CompletenessPercentage	Percentage of completed records.	String

Node	Description	Type
NumofInCompleteRecords	Number of completed records.	String
TotalRecords	Total records run.	String
ErrorMessage	Specific error message.	String

Request Message Description: getSourceCompleteComplianceInfo

For a description of this request message, see the following.

Node	Description	Type
startDateTme	Start date and time for a given job.	String
endDateTime	End date and time for a given job.	String
ObjectName	Business Object name. Either Contact, or Account.	String
SwiOrgSDHIO	An instance of the integration object, SwiOrgSDH.	Integration Object
SwiPerSDHIO	An instance of the integration object, SwiPerSDH.	Integration Object
SourceSystem	A registered source system contributing to changes in Oracle Customer Hub (UCM).	String

Response Message Description: getSourceCompleteComplianceInfo

For a description of this response message, see the following table.

Node	Description	Type
CompletenessPercentage	Percentage of completed records.	String
NumofInCompleteRecords	Number of completed records.	String
TotalRecords	Total records run.	String
ErrorMessage	Specific error message.	String

Request Message Description: getHubOrgIncompleteDetail

For a description of this request message, see the following table.

Node	Description	Type
endDateTime	End date and time for a given job.	String
startDateTime	Start date and time for a given job.	String
StartRowNum	The row number of the total number of records from which the records are to be given.	String
PageSize	Indicates the number of records to be returned.	String
SortSpec	The sort criteria applied to the business component.	String
NewQuery	Starts a new query.	String
SwiOrgIO	An instance of the integration object, SwiOrg.	Integration Object

Response Message Description: getHubOrgIncompleteDetail

For a description of this response message, see the following table.

Node	Description	Type
SwiOrgIO	An instance of the integration object, SwiOrgIO.	Integration Object
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
NumOutputObjects	Number of records being returned in this iteration of the request result set.	String
errorMessage	Specific error message.	String

Request Message Description: getHubPerIncompleteDetail

For a description of this request message, see the following table.

Node	Description	Type
endDateTime	End date and time for a given job.	String

Node	Description	Type
startDateTime	Start date and time for a given job.	String
StartRowNum	The row number of the total number of records from which the records are to be given.	String
PageSize	Indicates the number of records to be returned.	String
SortSpec	The sort criteria applied to the business component.	String
NewQuery	Starts a new query.	String
SwiPerIO	An instance of the integration object, SwiPer.	Integration Object

Response Message Description: getHubPerIncompleteDetail

For a description of this response message, see the following table.

Node	Description	Type
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
NumOutputObjects	Number of records being returned in this iteration of the request result set.	String
SwiPerIO	An instance of the integration object SwiPer.	Integration Object

Request Message Description: getSrcOrgIncompleteDetail

For a description of this request message, see the following table.

Node	Description	Type
endDateTime	End date and time for a given job.	String
startDateTime	Start date and time for a given job.	String
StartRowNum	The row number of the total number of records from which the records are to be given.	String
PageSize	Indicates the number of records to be returned.	String

Node	Description	Type
SortSpec	The sort criteria applied to the business component.	String
NewQuery	Starts a new query.	String
SwiOrgSDHIO	An instance of the integration object SwiOrgSDH.	Integration Object

Response Message Description: getSrcOrgIncompleteDetail

For a description of this response message, see the following table.

Node	Description	Type
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
NumOutputObjects	Number of output integration objects	String
SwiOrgSDHIO	An instance of the integration object, SwiOrgSDH.	Integration Object

Request Message Description: getSrcPerIncompleteDetail

For a description of this request message, see the following table.

Node	Description	Type
endDateTime	End date and time for a given job.	String
startDateTime	Start date and time for a given job.	String
StartRowNum	The row number of the total number of records from which the records are to be given.	String
PageSize	Indicates the number of records to be returned.	String
SortSpec	The sort criteria applied to the business component.	String
NewQuery	Starts a new query.	String
SwiPerSDHIO	An instance of the integration object SwiPerSDH.	Integration Object

Response Message Description: getSrcPerIncompleteDetail

For a description of this response message, see the following table.

Node	Description	Type
SwiOrgSDHIO	An instance of the integration object, SwiOrgSDH.	Integration Object
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
NumOutputObjects	Number of records being returned in this iteration of the request result set.	String
errorMessage	Specific error message.	String

CleanseCompletenessService Application Interface

This topic describes the application objects called by this Web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
UCM Data Governance Manager	Business Service	CSSSwiUCMDGMgrService

Data Object (Integration Object)

For a description of the data objects for this web service, see the following table.

Siebel Repository Name	External Name
SwiOrgIO	Account
SwiPerIO	Contact
SwiOrgSDHIO	UCM Account Source Data and History
SwiPerSDHIO	UCM Contact Source Data and History

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
getHubCompleteComplianceInfo	getHubCompleteComplianceInfo
getSourceCompleteComplianceInfo	getSourceCompleteComplianceInfo
getSystemsRegistrationInfo	getSystemsRegistrationInfo
getHubOrgIncompleteDetail	getHubOrgIncompleteDetail
getHubPerIncompleteDetail	getHubPerIncompleteDetail
getSrcOrgIncompleteDetail	getSrcOrgIncompleteDetail
getSrcPerIncompleteDetail	getSrcPerIncompleteDetail

GovernService

This Web service queries all the Account and Contact suspect records in the base table. It provides a means do the following:

- Approve suspect records to resolve nonsuspect records.
- Submit merge requests for suspect records that must be merged.

GovernService Operations

For a list of the operations associated with this web service, see the following table.

Name	Description
getOrganizationSuspectRecords	Queries all account records marked Suspect in the base table.
getPersonSuspectRecords	Queries all contact records marked Suspect in the base table.
removePendingRecord	Removes the pending flag for the account or contact record marked as Suspect in the base table.
submitMergeRequestOrganizationRecord	Submits a merge request for the account records in the base table.

Name	Description
submitMergeRequestPersonRecord	Submits a merge request for the contact records in the base table.

Request Message Description: getOrganizationSuspectRecords

For a description of this request message, see the following table.

Node	Description	Type
endDateTime	End date and time for a given job.	String
startDateTime	Start date and time for a given job.	String
StartRowNum	The row number of the total number of records from which the records are to be given.	String
PageSize	Indicates the number of records to be returned.	String
SortSpec	The sort criteria applied to the business component.	String
NewQuery	Starts a new query.	String

Response Message Description: getOrganizationSuspectRecords

For a description of this response message, see the following table.

Node	Description	Type
SwiOrgDeDupResultsIO	An instance of the integration object, SwiOrgDeDupResults.	Integration Object
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
NumOutputObjects	Number of records being returned in this iteration of the request result set.	String
ErrorCode	Error code (if any) that must be passed for the job.	String
ErrorMessage	Specific error message.	String

Request Message Description: getPersonSuspectRecords

For a description of this request message, see the following table.

Node	Description	Type
endDateTime	End date and time for a given job.	String
startDateTime	Start date and time for a given job.	String
StartRowNum	The row number of the total number of records from which the records are to be given.	String
PageSize	Indicates the number of records to be returned.	String
SortSpec	The sort criteria applied to the business component.	String
NewQuery	Starts a new query.	String

Response Message Description: getPersonSuspectRecords

For a description of this response message, see the following table.

Node	Description	Type
SwiPerDeDupResultsIO	An instance of the integration object, SwiPerDeDupResults.	Integration Object
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
NumOutputObjects	Number of records being returned in this iteration of the request result set.	String
ErrorCode	Error code (if any) that must be passed for the job.	String
ErrorMessage	Specific error message.	String

Request Message Description: removePendingRecord

For a description of this request message, see the following table.

Node	Description	Type
Id	Record ID.	String
ObjectName	Business object name, which is one of the following: contact or account.	String

Node	Description	Type
PendingStatus	The pending status of the record.	String

Response Message Description: removePendingRecord

For a description of this response message, see the following table.

Node	Description	Type
Status	Status of the record.	String
ErrorCode	Error code (if any) that must be passed for the job.	String
ErrorMessage	Specific error message.	String

Request Message Description: submitMergeRequestOrganizationRecord

For a description of this request message, see the following table.

Node	Description	Type
MatchId	Row ID of the duplicate record which is to be merged with the Master record.	String
MasterId	Row ID of the Master record for which duplicates have been found.	String

Response Message Description: submitMergeRequestOrganizationRecord

For a description of this response message, see the following table.

Node	Description	Type
ResponseMessage (UCMMergeRequestRs)	Response message to the UCM Merge request.	String
ErrorCode	Error code (if any) that must be passed for the job.	String
ErrorMessage	Specific error message.	String

Request Message Description: submitMergeRequestPersonRecord

For a description of this request message, see the following table.

Node	Description	Type
MatchId	Match ID.	Not applicable
MasterId	Master ID.	Not applicable

Response Message Description: submitMergeRequestPersonRecord

For a description of this response message, see the following table.

Node	Description	Type
ResponseMessage (UCMMergeRequestRs)	Response message to the UCM Merge request.	String.
ErrorCode	Error code (if any) that must be passed for the job.	String
ErrorMessage	Specific error message.	String

GovernService Application Interface

This topic describes the application objects called by this web service. For more information on application interfaces, refer to your application development documentation on Oracle Help Center.

Service Object (Business Service or Workflow)

For a description of the service objects for this web service, see the following table.

Siebel Repository Name	Boundary Object Type	Class
UCM Data Governance Manager	Business Service	CSSSwiUCMDGMgrService

Methods

For a description of the methods for this Web service, see the following table.

Operation	Method
getOrganizationSuspectRecords	getOrganizationSuspectRecords
getPersonSuspectRecords	getPersonSuspectRecords
removePendingRecord	removePendingRecord
submitMergeRequestOrganizationRecord	submitMergeRequestOrganizationRecord
submitMergeRequestPersonRecord	submitMergeRequestPersonRecord

21 Oracle Customer Hub (UCM) REST APIs Reference

Oracle Customer Hub (UCM) REST APIs Reference

This chapter provides reference information for Oracle Customer Hub (UCM) REST APIs. It includes the following topics:

- *About Oracle Customer Hub (UCM) REST APIs*
- *Provide Access to Siebel Workflow Processes and Siebel Business Services*
- *Get REST APIs Document in Open API Format*
- *Override Default Values in Siebel Workflow Process Properties*
- *Organization Service*
- *Person Service*
- *Group Service*
- *Financial Asset Service*
- *Merge Request Service*
- *UCM REST Publish Service*

About Oracle Customer Hub (UCM) REST APIs

UCM processes are build using Siebel Workflow Processes and Siebel Business Services. You can use Siebel REST Interfaces to invoke the workflows or execute the business services. Siebel REST Interface supports both JSON and XML format. For more information refer to *Siebel REST API Guide.*

Generalized format of HTTP POST request to run Siebel Workflow Process via REST Interface:

Request URL:

http://<Server Name>:<port>/Siebel/v1.0/**workflow**/<WF Process Name>

Request Headers:

Content-Type: application/json (or application/xml)

Authorization: <authorization info>

Request body in JSON format:

```
{
 "WFProcessProp1": "Value1",
 "WFProcessProp2": "Value2",
 //For hierarchy objects, such as "Siebel Message", pass it as child properties
}
```

Request body in XML format:

```
<?xml version="1.0" ?>
<request>
 <WFProcessProp1>Value1</WFProcessProp1>
 <WFProcessProp2>Value1</WFProcessProp2>
 //Complex hierarchy such as "Siebel Message" if any as a child
</request>
```

Generalized format of HTTP POST request to execute Siebel Business Service via REST Interface:

Request URL:

http://<Server Name>:<port>/Siebel/v1.0/**service/**<Business Service Name>/<Business Service Method Name>

Request Headers:

Content-Type: application/json (or application/xml)

Authorization: <authorization info>

Request body in JSON format:

```
{
 "BSMethodParam1": "Value1",
 "BSMethodParam2": "Value2",
 //For hierarchy objects, such as "Siebel Message", pass it as child properties
}
```

Request body in XML format:

```
<?xml version="1.0" ?>
<request>
 <BSMethodParam1>Value1</BSMethodParam1>
 <BSMethodParam2>Value1</BSMethodParam2>
 //Complex hierarchy such as "Siebel Message" if any as a child
</request>
```

Provide Access to Siebel Workflow Processes and Siebel Business Services

For a Siebel Workflow Process or a Siebel Business Service to be accessed via REST, administrator must assign it a Responsibility, as describe in the steps below:

Provide Access for Siebel Workflow Process

1. Navigate to **Administration – Application** > **Business Process Access** > **Access By Responsibility**
2. In the top applet *Business Process*, create a new record and pick a Business Process Name you want provide REST access.
3. In the bottom applet *Responsibility*, pick the value for the *Responsibility* field (example *Siebel Administrator* responsibility).
4. Click **Clear Cache** button.

Provide Access for Siebel Business Service

1. Navigate to **Administration – Application** > **Business Service Access** > **Access By Responsibility**.

2. In the top applet *Business Service*, create a new record and pick a Business Service Name you want provide REST access.
3. In the middle applet *Responsibility*, pick the value for the *Responsibility* field (example *Siebel Administrator* responsibility).
4. In the bottom applet *Business Service Method*, pick the Business Service Method Name.
5. Click **Clear Cache** button.

Get REST APIs Document in Open API Format

Use *describe* feature to get detail information of specific Siebel Workflow Process or Business Service, such as its input and output parameters in Open API format, including complex hierarchy structure like *Siebel Message*.

Format of HTTP GET request for Siebel Workflow Process REST Interface:

Request URL:

```
http://<Server Name>:<port>/Siebel/v1.0/workflow/<WF Process Name>/describe
```

Note: This call only returns Workflow Process Properties with In/Out type.

Format of HTTP GET request for Siebel Business Service Method REST Interface:

Request URL:

```
http://<Server Name>:<port>/Siebel/v1.0/service/<Business Service Name>/<Business Service Method Name>/
describe
```

Override Default Values in Siebel Workflow Process Properties

When invoking Siebel Workflow Process via REST Interface, calling application can override the default values specified in the WF Process Props, by sending in the new values in the REST request body. Note that only WF Process Props with In/Out type can be overridden.

Organization Service

Use this REST Interface to perform all operations related to the organization or the account object. This REST Interface is used for all organization operations, including insert, query, update, and delete operations to maintain organization information.

Organization Service Operations

For a list of the operations associated with this REST Interface, see the following table.

Name	Description
createOrganization	Creates an organization record.
getOrganization	Retrieves organization records, based on a user query.
deleteOrganization	Deletes an organization record.
updateOrganization	Updates an existing organization record.

The UCM REST Interface for Organization Service invokes the same workflow process used by UCM Organization Web Service, so it requires as input the same instance of *SwiOrganizationIO* integration object. Refer to the Organization Web Services section of this guide for detail description of this integration object.

Workflow Processes invoked by Organization Service:

Name	Workflow Process Name
createOrganization	UCM Organization Customer Profile Integration SOAP Process
getOrganization	UCM Organization Customer Profile Integration SOAP Query Process
deleteOrganization	UCM Organization Customer Profile Integration SOAP Process
updateOrganization	UCM Organization Customer Profile Integration SOAP Process

Sample Request – createOrganization

HTTP POST Request URL:

```
http://<Server Name>:<port>/Siebel/v1.0/workflow/UCM Organization Customer Profile Integration SOAP Process
```

Sample Request body in JSON format:

```
{
 "RealtimePubSub": "true", //Override WF Proc Prop
 "SwiOrganizationIO": { //Organization IO
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "",
 "IntObjectName": "SwiOrganizationIO",
 "MessageType": "Integration Object",
 "ListOfSwiOrganizationIO": {
 "ExternalSystemId": "Siebel01",
 "Account": {
 "operation": "Insert", //Insert operation
 "Id": "SBL01_ABC_ACCT_0111",
 "Location": "0111",
 "Main Fax Number": "1111111111",
 "Main Phone Number": "1111111111",
 "Name": "ABC ACCOUNT",
 "ListOfAccount_Business Address": {
 "Account_Business Address": {
 "IsPrimaryMVG": "Y",
 "Address Id": "SBL01_BUS_ADDR_0002",
 "City": "San Francisco",
 "Country": "USA",
 "Postal Code": "94104",
 "State": "CA",
 "Street Address": "222 Pine St"
 }
 }
```

```
 }
 }
 }
}
```

Sample Request body in XML format:

```
<?xml version="1.0" ?>
<request>
 <RealtimePubSub>true</RealtimePubSub> //Override WF Proc Prop
 <SwiOrganizationIO //Organization IO
 IntObjectFormat="Siebel Hierarchical"
 MessageId=""
 IntObjectName="SwiOrganizationIO"
 MessageType="Integration Object">
 <ListOfSwiOrganizationIO ExternalSystemId="Siebel01">
 <Account operation="Insert"> //Insert operation
 <Id>SBL01_ABC_ACCT_0111</Id>
 <Location>0111</Location>
 <Name>ABC Account</Name>
 <Main_spcPhone_spcNumber>1111111111</Main_spcPhone_spcNumber>
 </Account>
 </ListOfSwiOrganizationIO>
 </SwiOrganizationIO>
</request>
```

Person Service

Use this REST Interface to perform all operations related to the person or the contact object. This REST Interface is used for all person operations, including insert, query, update, and delete operations to maintain person information.

Person Service Operations

For a list of the operations associated with this REST Interface, see the following table.

Name	Description
createPerson	Creates a person record.
getPerson	Retrieves person records, based on a user query.
deletePerson	Deletes a person record.
updatePerson	Updates an existing person record.

The UCM REST Interface for Person Service invokes the same workflow process used by UCM Person Web Service, so it requires as input the same instance of *SwiPersonIO* integration object. Refer to the Person Web Services section of this guide for detail description of this integration object.

Workflow Processes invoked by Person Service:

Name	Workflow Process Name
createPerson	UCM Person Customer Profile Integration SOAP Process
getPerson	UCM Person Customer Profile Integration SOAP Query Process
deletePerson	UCM Person Customer Profile Integration SOAP Process
updatePerson	UCM Person Customer Profile Integration SOAP Process

Sample Request - getPerson

HTTP POST Request URL:

```
http://<Server Name>:<port>/Siebel/v1.0/workflow/UCM Person Customer Profile Integration SOAP Query Process
```

Sample Request body in JSON format:

```
{
 "SwiPersonIO": { //Person IO
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "",
 "IntObjectName": "SwiPersonIO",
 "MessageType": "Integration Object",
 "ListOfSwiPersonIO": {
 "ExternalSystemId": "Siebel01",
 "Contact": {
 "operation": "Query", //Query operation
 "First Name": "JERRY",
 "Last Name": "RICE",
 "Work Phone #": "1111111111",
 "Home Phone #": "1111111111",
 "Email Address": "jerry.rice@gmail.com"
 }
 }
 }
}
```

Sample Request body in XML format:

```
<?xml version="1.0" ?>
<request>
 <SwiPersonIO //Person IO
 IntObjectFormat="Siebel Hierarchical"
 MessageId=""
 IntObjectName="SwiPersonIO"
 MessageType="Integration Object">
 <ListOfSwiPersonIO ExternalSystemId="Siebel01">
 <Contact operation="Query"> //Query operation
 <First_spcName>JERRY</First_spcName>
 <Last_spcName>RICE</Last_spcName>
 <Work_spcPhone_spc_pnd>1111111111</Work_spcPhone_spc_pnd>
 <Email_spcAddress>jerry.rice@gmail.com</Email_spcAddress>
 </Contact>
 </ListOfSwiPersonIO>
 </SwiPersonIO>
</request>
```

Group Service

Use this REST Interface to perform all operations related to the group or the household object. This REST Interface is used for all group operations, including insert, query, update, and delete operations to maintain group information.

Group Service Operations

For a list of the operations associated with this REST Interface, see the following table.

Name	Description
createGroup	Creates a group record.
getGroup	Retrieves group records, based on a user query.
deleteGroup	Deletes a group record.
updateGroup	Updates an existing group record.

The UCM REST Interface for Group Service invokes the same workflow process used by UCM Group Web Service, so it requires as input the same instance of *SwiGroupIO* integration object. Refer to the Group Web Services section of this guide for detail description of this integration object.

Workflow Processes invoked by Group Service:

Name	Workflow Process Name
createGroup	UCM Group Customer Profile Integration SOAP Process
getGroup	UCM Group Customer Profile Integration SOAP Query Process
deleteGroup	UCM Group Customer Profile Integration SOAP Process
updateGroup	UCM Group Customer Profile Integration SOAP Process

Sample Request - deleteGroup

HTTP POST Request URL:

```
http://<Server Name>:<port>/Siebel/v1.0/workflow/UCM Group Customer Profile Integration SOAP Process
```

Sample Request body in JSON format:

```
{
 "SwiGroupIO": { //Group IO
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "",
 "IntObjectName": "SwiGroupIO",
 "MessageType": "Integration Object",
 "ListOfSwiGroupIO": {
 "ExternalSystemId": "Siebel01",
 "Household": {
 "operation": "Delete", //Delete operation
 "Household Name": "ABC HOUSEHOLD",
```

```
"Household Number": "ABC-0002",
"Phone Number": "2222222222",
"Fax Number": "2222222222"
}
}
}
}
```

Sample Request body in XML format:

```
<?xml version="1.0" ?>
<request>
 <SwiGroupIO //Group IO
 IntObjectFormat="Siebel Hierarchical"
 MessageId=""
 IntObjectName="SwiGroupIO"
 MessageType="Integration Object">
 <ListOfSwiGroupIO ExternalSystemId="Siebel01">
 <Household operation="Delete"> //Delete operation
 <Household_spcName>ABC HOUSEHOLD</Household_spcName>
 <Household_spcNumber>ABC-0002</Household_spcNumber>
 <Phone_spcNumber>2222222222</Phone_spcNumber>
 <Fax_spcNumber>2222222222</Fax_spcNumber>
 </Household>
 </ListOfSwiGroupIO>
 </SwiGroupIO>
</request>
```

Financial Asset Service

Use this REST Interface to perform all operations related to the financial asset or the financial account object. This REST Interface is used for all financial asset operations, including insert, query, update, and delete operations to maintain financial asset information.

Financial Asset Service Operations

For a list of the operations associated with this REST Interface, see the following table.

Name	Description
createFinancialAsset	Creates a financial asset record.
getFinancialAsset	Retrieves financial asset records, based on a user query.
deleteFinancialAsset	Deletes a financial asset record.
updateFinancialAsset	Updates an existing financial asset record.

The UCM REST Interface for Financial Asset Service invokes the same workflow process used by UCM Financial Asset Web Service, so it requires as input the same instance of *SwiFinancialAssetIO* integration object. Refer to the Financial Asset Web Services section of this guide for detail description of this integration object.

Workflow Processes invoked by Financial Asset Service:

Name	Workflow Process Name
createFinancialAsset	UCM Financial Asset Customer Profile Integration SOAP Process
getFinancialAsset	UCM Financial Asset Customer Profile Integration SOAP Query Process
deleteFinancialAsset	UCM Financial Asset Customer Profile Integration SOAP Process
updateFinancialAsset	UCM Financial Asset Customer Profile Integration SOAP Process

Sample Request - updateFinancialAsset

HTTP POST Request URL:

```
http://<Server Name>:<port>/Siebel/v1.0/workflow/UCM Financial Asset Customer Profile Integration SOAP
 Process
```

Sample Request body in JSON format:

```
{
 "RealtimePubSub": "true", //Override WF Proc Prop
 "SwiFinancialAssetIO": { //Financial Asset IO
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "",
 "IntObjectName": "SwiFinancialAssetIO",
 "MessageType": "Integration Object",
 "ListOfSwiFinancialAssetIO": {
 "ExternalSystemId": "Siebel01",
 "FINCORP Account": [
 {
 "operation": "Update", //Update operation
 "Id": "SBL01_ABC_ACCT_0001",
 "Account Name": "ABC ACCOUNT",
 "Account Number": "ABC-0001",
 "Current Balance": "5000",
 "Product Id": "1+KNC+102",
 "Name": "0000894 1+KNC+102",
 "Start Date": "09/01/2010 00:00:00",
 "ListOfContact": {
 "Contact": [
 {
 "Id": "SBL01_JOE_MONTANA",
 "Last Name": "MONTANA",
 "First Name": "JOE",
 "Home Phone #": "1234567890",
 "Work Phone #": "1234567890"
 }
 ]
 }
 }
 ]
 }
 }
}
```

Sample Request body in XML format:

```
<?xml version="1.0" ?>
<request>
 <RealtimePubSub>true</RealtimePubSub> //Override WF Proc Prop
 <SwiFinancialAssetIO //Financial Asset IO
 IntObjectFormat="Siebel Hierarchical"
 MessageId=""
 IntObjectName="SwiFinancialAssetIO"
```

```
MessageType="Integration Object">
<ListOfSwiFinancialAssetIO ExternalSystemId="Siebel01">
<FINCORP_spcAccount operation="Update"> //Update operation
<Id>SBL01_ABC_ACCT_0001</Id>
<Account_spcName>ABC ACCOUNT</Account_spcName>
<Account_spcNumber>ABC-0111</Account_spcNumber>
<Current_spcBalance>5000</Current_spcBalance>
<Product_spcId>1+KNC+102</Product_spcId>
<Name>0000894 1+KNC+102</Name>
<Start_spcDate>09/01/2010 00:00:00</Start_spcDate>
<ListOfContact>
<Contact>
<Id>SBL01_JOE_MONTANA</Id>
<Last_spcName>MONTANA</Last_spcName>
<First_spcName>JOE</First_spcName>
</Contact>
</ListOfContact>
</FINCORP_spcAccount>
</ListOfSwiFinancialAssetIO>
</SwiFinancialAssetIO>
</request>
```

Merge Request Service

This REST Interface is used for creating merge request operations, including submit merge request for Organization and Person.

Merge Request Service Operations

For a list of the operations associated with this REST Interface, see the following table.

Name	Description
submitOrganizationMergeRequest	Creates an organization merge request record.
submitPersonMergeRequest	Creates a person merge request record.

The UCM REST Interface for Merge Request Service invokes the business service *UCM Merge Service*, method *SubmitMergeRequest*. Below table lists the input arguments for this business service method.

SubmitMergeRequest method input arguments:

Name	Data Type	Optional	Comment
MasterId	String		Id of surviving record after the merge operation.
MatchId	String		Id of matching record to be merged.
ObjectType	String		Support merge request for Account or Contact.
RequestStatus	String		Status of matching record.

Name	Data Type	Optional	Comment
SystemId	String	Yes	External System Id of matching record.

Sample Request - submitOrganizationMergeRequest

HTTP POST Request URL:

```
http://<Server Name>:<port>/Siebel/v1.0/service/UCM Merge Service/SubmitMergeRequest
```

Sample Request body in JSON format:

```
{
 "body": {
 "RequestStatus": "Pending Review",
 "SystemId": "Siebel01",
 "MatchId": "SBL01_ACCT_ABC",
 "MasterId": "SBL01_ACCT_XYZ",
 "ObjectType": "Account"
 }
}
```

Sample Request body in XML format:

```
<?xml version="1.0" ?>
<request>
 <RequestStatus>Pending Review</RequestStatus>
 <SystemId>Siebel01</SystemId>
 <MatchId>SBL01_ACCT_ABC</MatchId>
 <MasterId>SBL01_ACCT_XYZ</MasterId>
 <ObjectType>Account</ObjectType>
</request>
```

UCM REST Publish Service

UCM can publish message in XML or JSON format, based on *Message Type* specified for each registered system. For systems with SOAP *Message Type*, UCM publishes the message in XML format; for systems with JSON *Message Type*, UCM publishes the message in JSON format.

In addition, you can configure a Middleware system with PROXY protocol, to publish message by invoking a business service method, and use Outbound REST Service `UCMRESTPublish` to publish message in JSON format.

To Configure Outbound REST Services

1. Navigate to **Administration – Web Services** > **Outbound REST Services**.
2. In the top applet **Outbound REST Services**, select `UCMRESTPublish` (if not exists, click on the **Import** button and import it from XML file.
3. In the bottom applet *Service Methods* modify the *URL Address* to point to your server and port number where the publish message will be sent.

To Configure Middleware System to Use Outbound REST Publish Service

1. Navigate to **Administration – Universal Customer Master** > **System Registrations**.

2. Create a new system, check the *Middleware System* flag, select PROXY as *Protocol Type*, and JSON as *Message Type*.
3. Drill-down on the **System Id** and select *Publish / Subscribe* tab in the bottom applet.
4. Create new record for each supported object (for example Account, Contact, FINCORD Account, or Household).
5. Pick a **Publish Frequency** value (example Real Time) and enter a valid **End Date**.
6. In the field **Business Service** select Outbound REST Service **UCMRESTPublish**.
7. In the field **Business Service Method** select the appropriate method for each object, see table below.

UCMRESTPublish has methods to publish JSON message for each supported object:

Object Name	Business Service Method
Account	publishOrganization
Contact	publishPerson
FINCORP Account	publishFinancialAsset
Household	publishGroup

The input argument for these publish methods is a REST Integration Object, one for each supported object:

Publish Method Name	REST Publish IO Name
publishOrganization	RestOrganizationPublishIO (based on SwiOrganizationPublishIO)
publishPerson	RestPersonPublishIO (based on SwiPersonPublishIO)
publishFinancialAsset	RestFinancialAssetPublishIO (based on SwiFinancialAssetPublishIO)
publishGroup	RestGroupPublishIO (based on SwiGroupPublishIO)

Note: If you want to publish additional child components or fields, add them to both REST and SWI Publish Integration Objects.

22 Schema Changes for Oracle Customer Hub (UCM) Version 8.2

Schema Changes for Oracle Customer Hub (UCM) Version 8.2

This chapter contains information about table column changes in the schema for Oracle Customer Hub (UCM) version 8.2. It contains the following topic:

- *Schema Changes for Oracle Customer Hub (UCM) for Siebel CRM 14.0*
- *Modified Tables for Oracle Customer Hub (UCM) for Siebel CRM 14.0*
- *New Tables for Oracle Customer Hub (UCM) Version 8.2*
- *Modified Tables for Oracle Customer Hub (UCM) Version 8.2*
- *Modified Tables for Oracle Customer Hub (UCM) Version 8.2*

Schema Changes for Oracle Customer Hub (UCM) for Siebel CRM 14.0

The following topics list schema changes made as part of Oracle Customer Hub (UCM) for Siebel CRM 14.0.

- *New Tables for Oracle Customer Hub (UCM) for Siebel CRM 14.0*
- *Modified Tables for Oracle Customer Hub (UCM) for Siebel CRM 14.0*

New Tables for Oracle Customer Hub (UCM) for Siebel CRM 14.0

The following new tables were added as part of Oracle Customer Hub (UCM) for Siebel CRM 14.0.

- *S_CONSMPROF_UPD Table*
- *S_UCM_CN_SMPROF Table*
- *S_UCM_CN_SMATTR Table*
- *S_CIF_SMPROFMAP Table*
- *S_CIF_SMATTRMAP Table*

S_CONSMPROF_UPD Table

The following table shows details for the new S_CONSMPROF_UPD table which is used to configure survivorship attribute settings for Contact Social Media Profile entity records.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Required. Siebel system field.
CON_SM_PROF_ID	Varchar (15)	Required. The child entity row ID.
FLD_VAL_RULE_ID	Varchar (15)	Field value row ID.
EXT_SYS_ID	Varchar (15)	Required. External system ID.
LAST_MOD_TS	UTC Date Time (7)	Last modified date.
CONFLICT_ID	Varchar (15)	Required. Siebel system field.
CREATED	UTC Date Time (7)	Required. Siebel system field.
CREATED_BY	Varchar (15)	Required. Siebel system field.
DB_LAST_UPD	UTC Date Time (7)	Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Required. Siebel system field.
LAST_UPD_BY	Varchar (15)	Required. Siebel system field.
MODIFICATION_NUM	Number (22)	Required. Siebel system field.

S_UCM_CN_SMPROF Table

The following table shows details for the new S_UCM_CN_SMPROF source data history (SDH) table which is used to provide support for Social Media Profile data.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Siebel system field.

Column Name	Data Type	Comments
CONFLICT_ID	Varchar (15)	Siebel system field.
CREATED	UTC Date Time (7)	Siebel system field.
CREATED_BY	Varchar (15)	Siebel system field.
DB_LAST_UPD	UTC Date Time (7)	Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Siebel system field.
LAST_UPD_BY	Varchar (15)	Siebel system field.
MODIFICATION_NUM	Number (22)	Siebel system field.
BV_LAST_UPD	UTC Date Time (7)	the last update date time of the best version record.
BV_LAST_UPD_BY	Varchar (15)	The last updated user login of the best version record.
EXT_SYST_ID	Varchar (15)	External system ID.
UCM_EXT_ID	Varchar (100)	The UCM external unique identifier.
UCM_EXT_LOGIN	Varchar (30)	External user login.
UCM_EXT_UPD_TS	UTC Date Time (7)	The last updated date of the external system.
UCM_TXN_DESC	Varchar (250)	The UCM transaction description.
UCM_TYPE_CD	Varchar (30)	The status or type of the record.
UCM_UID	Varchar (100)	The UCM unique identifier.
CON_SM_PROF_ID	Varchar (15)	The best version ID. A null value means the child record is unlinked.
UCM_CONTACT_ID	Varchar (15)	The parent UCM contact ID.

Column Name	Data Type	Comments
CONTACT_ID	Varchar (15)	The BV parent contact ID.
AUTHOR1_ID_VAL	Varchar (255)	The author ID value provided by a data provider or a social integrator.
AUTHOR_LINK_URL	Varchar (500)	The unique URL of the author profile on a social media site.
AUTHOR_NAME	Varchar (100)	The unique user name of the contact on a specific social media site (such as Twitter or Facebook). For example: @user_name. This author name is unique within the social media site.
CHANNEL_CD	Varchar (30)	A social media channel such as Facebook or Twitter which is associated with a contact or person.
SUPPRESS_PROF_FLG	Character(1)	The flag indicates whether social profile attributes for a specific profile or author from a given social media channel are retrieved.

S_UCM_CN_SMATTR Table

The following table shows details for the new S_UCM_CN_SMATTR source data history (SDH) table which is used to provide support for Social Media Attribute data.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Siebel system field.
CONFLICT_ID	Varchar (15)	Siebel system field.
CREATED	UTC Date Time (7)	Siebel system field.
CREATED_BY	Varchar (15)	Siebel system field.
DB_LAST_UPD	UTC Date Time (7)	Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Siebel system field.

Column Name	Data Type	Comments
LAST_UPD_BY	Varchar (15)	Siebel system field.
MODIFICATION_NUM	Number (22)	Siebel system field.
BV_LAST_UPD	UTC Date Time (7)	The last update date time of the best version record.
BV_LAST_UPD_BY	Varchar (15)	The last updated user login of the best version record.
EXT_SYST_ID	Varchar (15)	External system ID.
UCM_EXT_ID	Varchar (100)	The UCM external unique identifier.
UCM_EXT_LOGIN	Varchar (30)	External user login.
UCM_EXT_UPD_TS	UTC Date Time (7)	The last updated date of the external system.
UCM_TXN_DESC	Varchar (250)	The UCM transaction description.
UCM_TYPE_CD	Varchar (30)	The status or type of the record.
UCM_UID	Varchar (100)	The UCM unique identifier.
CON_SM_ATTR_ID	Varchar (15)	The best version ID. A null value means the child record is unlinked.
UCM_CON_SM_PROF_ID	Varchar (15)	The parent UCM contact social media profile ID.
CON_SM_PROF_ID	Varchar (15)	The BV parent contact social media profile ID.
NAME	Varchar (100)	Attribute name of the profile recieved from a source system, such as: Follower, Tweets, Friends, Likes, Age, DoB, and so on.
SOURCE_CD	Varchar (30)	The source (data provider) through which the profile data, such as: Buzzient, Visible, Actionly, Hotsuite, and so on.

Column Name	Data Type	Comments
VALUE	Varchar (500)	Value for an attribute. For example, Name = Pradeep, Followers = 59, Friends = 150, and Likes = 2.

S_CIF_SMPROFMAP Table

The following table shows details for the new S_CIF_SMPROFMAP cross reference table which is used to provide support for Social Media Profile Cross Reference data.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Siebel system field.
CON_SMPROF_ID	Varchar 15)	Social Media Profile Best Version Record ID
CIF_EXT_SYST_ID	Varchar (15)	External System ID.
EXT_CON_SMPROF_ID1	Varchar (100)	First Cross Reference ID.
EXT_CON_SMPROF_ID2	Varchar (100)	Place holder for second cross reference ID.
EXT_CON_SMPROF_ID3	Varchar (100)	Place holder for third cross reference ID.
COMMENTS	Varchar (250)	Comments.
CONFLICT_ID	Varchar (15)	Siebel system field.
CREATED	UTC Date Time (7)	Siebel system field.
CREATED_BY	Varchar (15)	Siebel system field.
DB_LAST_UPD	UTC Date Time (7)	Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Siebel system field.
LAST_UPD_BY	Varchar (15)	Siebel system field.

Column Name	Data Type	Comments
MODIFICATION_NUM	Number (22)	Siebel system field.

S_CIF_SMATTRMAP Table

The following table shows details for the new S_CIF_SMATTRMAP cross reference table which is used to provide support for Social Media Profile Attribute Cross Reference data.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Siebel system field.
CON_SMATTR_ID	Varchar 15)	Social Media Profile Attribute Best Version Record ID.
CIF_EXT_SYST_ID	Varchar (15)	External System ID.
EXT_CON_SMATTR_ID1	Varchar (100)	First Cross Reference ID.
EXT_CON_SMATTR_ID2	Varchar (100)	Place holder for second cross reference ID.
EXT_CON_SMATTR_ID3	Varchar (100)	Place holder for third cross reference ID.
COMMENTS	Varchar (250)	Comments.
CONFLICT_ID	Varchar (15)	Siebel system field.
CREATED	UTC Date Time (7)	Siebel system field.
CREATED_BY	Varchar (15)	Siebel system field.
DB_LAST_UPD	UTC Date Time (7)	Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Siebel system field.
LAST_UPD_BY	Varchar (15)	Siebel system field.

Column Name	Data Type	Comments
MODIFICATION_NUM	Number (22)	Siebel system field.

Modified Tables for Oracle Customer Hub (UCM) for Siebel CRM 14.0

The following table was modified as part of Oracle Customer Hub (UCM) for Siebel CRM 14.0.

S_CON_SM_PROF Table

The following table shows details of the S_CON_SM_PROF table which is used for Contact Social Media Profile entity records.

Column Name	Data Type	Comments
START_DT	UTC Date Time (7)	Not required for child survivorship, but it is recommended you use this field if you are using the End Date field.
END_DT	UTC Date Time (7)	Used for survivorship to indicate an obsolete Contact Social Media Profile.

Schema Changes for Oracle Customer Hub (UCM) Version 8.2

The following topics list schema changes made as part of Oracle Customer Hub (UCM) version 8.2 (for Siebel CRM 8.1.1.10).

- *New Tables for Oracle Customer Hub (UCM) Version 8.2*
- *Modified Tables for Oracle Customer Hub (UCM) Version 8.2*

New Tables for Oracle Customer Hub (UCM) Version 8.2

The following new tables were added as part of Oracle Customer Hub (UCM) version 8.2 (for Siebel CRM 8.1.1.7 Quick Fix 07AN).

- *S_CIF_ADDR_MAP Table*

- *S_CIF_CON_NM_MP Table*
- *S_CIF_CM_ADR_MP Table*
- *S_FLD_VAL_RULE Table*
- *S_FLD_CONF_LVL Table*
- *S_COMM_ADDR_UPD Table*
- *S_CON_NAME_UPD Table*
- *S_CON_IDNTY_UPD Table*
- *S_ASSET_UPD Table*
- *S_ADDR_PER_UPD Table*
- *S_PARTY_REL_UPD Table*
- *S_CIF_SMATTRMAP Table*

S_CIF_ADDR_MAP Table

The following table shows details for the new S_CIF_ADDR_MAP table which is used for Address cross references.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Siebel system field.
ADDR_ID	Varchar (15)	Address best version record ID. The foreign key table is S_ADDR_PER.
CIF_EXT_SYST_ID	Varchar (15)	External system ID. The foreign key table is S_CIF_EXT_SYST.
EXT_ADDR_ID1	Varchar (100)	The first cross reference ID.
EXT_ADDR_ID2	Varchar (100)	Place holder for second cross reference ID.
EXT_ADDR_ID3	Varchar (100)	Place holder for third cross reference ID.
PAR_OBJTYPE_CD	Varchar (30)	Parent object type: Account, or Contact. The bounded LOV is :UCM_OBJECT_TYPE_CD.
COMMENTS	Varchar (250)	Comments.
CONFLICT_ID	Varchar (15)	Siebel system field.
CREATED	UTC Date Time (7)	Siebel system field.

Column Name	Data Type	Comments
CREATED_BY	Varchar (15)	Siebel system field.The foreign key table is S_USER.
DB_LAST_UPD	UTC Date Time (7)	Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Siebel system field.
LAST_UPD_BY	Varchar (15)	Siebel system field. The foreign key table is S_USER.
MODIFICATION_NUM	Number (22)	Siebel system field.

S_CIF_CON_NM_MP Table

The following table show details of the new S_CIF_CON_NM_MP table which is used for Name cross references.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Required. Siebel system field.
CON_NAME_ID	Varchar (15)	Required. The name of the best version record ID. The foreign key table is S_ CONTACT_NAME.
CIF_EXT_SYST_ID	Varchar (15)	Required. The external system ID. The foreign key table is S_CIF_EXT_SYST.
EXT_CON_NAME_ID1	Varchar (100)	Required. The first cross reference ID.
EXT_CON_NAME_ID2	Varchar (100)	Place holder for second cross reference ID.
EXT_CON_NAME_ID3	Varchar (100)	Place holder for third cross reference ID.
COMMENTS	Varchar (250)	Comments.
CONFLICT_ID	Varchar (15)	Required. Siebel system field.
CREATED	UTC Date Time (7)	Required. Siebel system field.

Column Name	Data Type	Comments
CREATED_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
DB_LAST_UPD	UTC Date Time (7)	Required. Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Required. Siebel system field.
LAST_UPD	UTC Date Time (7)	Siebel system field.
LAST_UPD_BY	Varchar (15)	Siebel system field. The foreign key table is S_USER.
MODIFICATION_NUM	Number (22)	Siebel system field.

S_CIF_CM_ADR_MP Table

The following table shows details of the new S_CIF_CM_ADR_MP which is used for Communication Address (Email and Phone) cross references.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Required. Siebel system field.
PER_COMM_ADDR_ID	Varchar (15)	Required. The communication address best version record ID. The foreign key table is S_PER_COMM_ADDR.
COMM_MEDIUM_CD	Varchar (30)	Required. Comm Medium Code. Comm Medium Code refers to the code for the communication media channel. Examples are email, ICQ, phone, and so on.
CIF_EXT_SYST_ID	Varchar (15)	Required. The external system ID. The foreign key table is S_CIF_EXT_SYST.
EXT_COMM_ADDR_ID1	Varchar (100)	Required. The first cross reference ID.
EXT_COMM_ADDR_ID2	Varchar (100)	The second cross reference ID.
EXT_COMM_ADDR_ID3	Varchar (100)	The third cross reference ID.
COMMENTS	Varchar (250)	Comments.

Column Name	Data Type	Comments
CONFLICT_ID	Varchar (15)	Required. Siebel system field.
CREATED	UTC Date Time (7)	Required. Siebel system field.
CREATED_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
DB_LAST_UPD	UTC Date Time (7)	Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Required. Siebel system field.
LAST_UPD_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
MODIFICATION_NUM	Number (22)	Required. Siebel system field.

S_FLD_VAL_RULE Table

The following table shows details for the new S_FLD_VAL_RULE table which contains field value rule definitions for all entities.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Required. Siebel system field.
SEQ_NUM	Number (22)	The sequence number to set priority as required.
NAME	Varchar (100)	Rule name.
RULE_SET_ID	Varchar (30)	Not required. The foreign key table is S_UCM_RULE_SET.
COMP_FIELD_NAME	Varchar (75)	Field name for the comparison value of Type in a business address.
COMP_FIELD_VALUE	Varchar (140)	The values of the comparison column Business Personal.

Column Name	Data Type	Comments
COMP_RULE_CD	Varchar (75)	The comparison rule used for Source, Recent, Master, and External records.
STATUS_CD	Varchar (75)	Status. Values are active or inactive.
BUS_SERVICE_NAME	Varchar (75)	The external rule business service.
METHOD_NAME	Varchar (75)	The external rule business service method.
RULE_MODULE_NAME	Varchar (150)	The external rule module.
DESC_TXT	Varchar (250)	Description.
CONFLICT_ID	Varchar(15)	Required. Siebel system field.
CREATED	UTC date time (7)	Required. Siebel system field.
CREATED_BY	Varchar (15)	Required. Siebel system field.
DB_LAST_UPD	UTC Date Time (7)	Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Required. Siebel system field.
LAST_UPD_BY	Varchar (15)	Required. Siebel system field.
MODIFICATION_NUM	Number (22)	Required. Siebel system field.

S_FLD_CONF_LVL Table

The following table shows details for the new S_FLD_CONF_LVL table which is used to set the child field rule confidence level in subscribed systems.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Required. Siebel system field.

Column Name	Data Type	Comments
FLD_VAL_RULE_ID	Varchar (15)	Required. Field value row ID. The foreign key table is S_FLD_VAL_ RULE.
EXT_SYS_ID	Varchar (15)	Required. The external system ID. The foreign key table is S_CIF_EXT_ SYS.
CONFIDENCE_LVL	Number (22)	Confidence level expressed in a percentage.
DESC_TXT	Varchar (250)	Description.
CONFLICT_ID	Varchar (15)	Required. Siebel system field.
CREATED	UTC Date Time (7)	Required. Siebel system field.
CREATED_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
DB_LAST_UPD	UTC Date Time (7)	Required. Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Required. Siebel system field.
LAST_UPD	UTC Date Time (7)	Siebel system field.
LAST_UPD_BY	Varchar (15)	Siebel system field. The foreign key table is S_USER.
MODIFICATION_NUM	Number (22)	Siebel system field.

S_COMM_ADDR_UPD Table

The following table shows details for the new S_COMM_ADDR_UPD table which is used to configure survivorship attribute settings for the email and alternate phone entities.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Required. Siebel system field.
COMM_ID	Varchar (15)	Required. The entity row ID for the email and alternate phone fields. The

Column Name	Data Type	Comments
		foreign key table is S_PER_COMM_ADDR.
FLD_VAL_RULE_ID	Varchar (15)	Field value row ID. The foreign key table is S_FKD_VAL_RULE.
EXT_SYS_ID	Varchar (15)	The external system ID. The foreign key table is S_CIF_EXT_SYS.
LAST_MOD_TS	UTC Date Time (7)	Last modified date time.
DESC_TXT	Varchar (250)	Description.
CONFLICT_ID	Varchar (15)	Required. Siebel system field.
CREATED	UTC Date Time (7)	Required. Siebel system field.
CREATED_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
DB_LAST_UPD	UTC Date Time (7)	Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Required. Siebel system field.
LAST_UPD_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
MODIFICATION_NUM	Number (22)	Required. Siebel system field.

S_CON_NAME_UPD Table

The following table shows details for the new S_CON_NAME_UPD table which is used to configure survivorship attribute settings for records associated with the name entity.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Required. Siebel system field.

Column Name	Data Type	Comments
NAME_ID	Varchar (15)	Required. The entity row ID for the name field. The foreign key table is S_CONTACT_NAME.
FLD_VAL_RULE_ID	Varchar (15)	The field value row ID. The foreign key table is S_FLD_VAL_RULE.
EXT_SYS_ID	Varchar (15)	The external system ID. The foreign key table is S_CIF_EXT_SYS.
LAST_MOD_TS	UTC Date Time (7)	The last modified date and time.
DESC_TXT	Varchar (250)	Description.
CONFLICT_ID	Varchar (15)	Required. Siebel system field.
CREATED	UTC Date Time (7)	Required. Siebel system field.
CREATED_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
DB_LAST_UPD	UTC Date Time (7)	Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Required. Siebel system field.
LAST_UPD_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
MODIFICATION_NUM	Number (22)	Required. Siebel system field.

S_CON_IDNTY_UPD Table

The following table shows details for the new S_CON_IDNTY_UPD table which is used to configure survivorship attribute settings for identification entity records.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Required. Siebel system field.

Column Name	Data Type	Comments
IDENT_ID	Varchar (15)	Required. The entity row ID for the name field. The foreign key table is S_CON_IDNTY_DOC.
FLD_VAL_RULE_ID	Varchar (15)	Field value row ID. The foreign key table is S_FLD_VAL_RULE.
EXT_SYS_ID	Varchar (15)	External system ID. The foreign key table is S_CIF_EXT_SYS.
LAST_MOD_TS	UTC Date Time (7)	The last modified date and time.
DESC_TXT	Varchar (250)	Description.
CONFLICT_ID	Varchar (15)	Required. Siebel system field.
CREATED	UTC Date Time (7)	Required. Siebel system field.
CREATED_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
DB_LAST_UPD	UTC Date Time (7)	Required. Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Siebel system field.
LAST_UPD_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
MODIFICATION_NUM	Number (22)	Required. Siebel system field.

S_ASSET_UPD Table

The following table shows details for the new S_ASSET_UPD table which is used to configure survivorship attribute settings for Financial Account entity records.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Required. Siebel system field.

Column Name	Data Type	Comments
ASSET_ID	Varchar (15)	Required. The entity row ID for the financial account entity. The foreign key table is S_ASSET.
FLD_VAL_RULE_ID	Varchar (15)	Field value row ID. The foreign key table is S_FLD_VAL_RULE.
PAR_CON_ID	Varchar (15)	The parent row ID for the Contact table. The foreign key table is S_CONTACT.
PAR_ACCNT_ID	Varchar (15)	The parent row ID for the Account table. The foreign key table is S_ORG_EXT.
PAR_ORGGRP_ID	Varchar (15)	The row ID of the Household table. The foreign key table is S_ORG_GROUP.
LAST_MOD_TS	UTC Date Time (7)	The last modified date and time.
DESC_TXT	Varchar (250)	Description.
CONFLICT_ID	Varchar (15)	Required. Siebel system field.
CREATED	UTC Date Time (7)	Required. Siebel system field.
CREATED_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
DB_LAST_UPD	UTC Date Time (7)	Required. Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Siebel system field.
LAST_UPD_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
MODIFICATION_NUM	Number (22)	Required. Siebel system field.

S_ADDR_PER_UPD Table

The following table shows details for the new S_ADDR_PER_UPD table which is used to configure survivorship attribute settings for Address entity records.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Required. Siebel system field.
ADDR_ID	Varchar (15)	Required. The row ID of the Address object. The foreign key table is S_ADDR_PER.
FLD_VAL_RULE_ID	Varchar (15)	The field value row ID. The foreign key table is S_FLD_VAL_RULE.
PAR_CON_ID	Varchar (15)	The parent row ID for the Contact table. The foreign key table is S_CONTACT.
PAR_ACCNT_ID	Varchar (15)	The parent row ID for the Account table. The foreign key table is S_ORG_EXT.
PAR_ORGGRP_ID	Varchar (15)	The row ID of the Household table. The foreign key table is S_ORG_GROUP.
PAR_ASSET_ID	Varchar (15)	The parent row ID of the FINCORP Account table. The foreign key table is S_ASSET.
EXT_SYS_ID	Varchar (15)	The external system ID. The foreign key table is S_CIF_EXT_SYS.
LAST_MOD_TS	UTC Date Time (7)	The last modified date and time.
DESC_TXT	Varchar (250)	Description.
CONFLICT_ID	Varchar (15)	Required. Siebel system field.
CREATED	UTC Date Time (7)	Required. Siebel system field.
CREATED_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.

Column Name	Data Type	Comments
DB_LAST_UPD	UTC Date Time (7)	Required. Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Siebel system field.
LAST_UPD_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
MODIFICATION_NUM	Number (22)	Required. Siebel system field.

S_PARTY_REL_UPD Table

The following table shows details for the new S_PARTY_REL_UPD table which is used to configure survivorship attribute settings for address entity records.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Required. Siebel system field.
ADDR_ID	Varchar (15)	Required. The row ID of the Address object. The foreign key table is S_ADDR_PER.
FLD_VAL_RULE_ID	Varchar (15)	The field value row ID. The foreign key table is S_FLD_VAL_RULE.
PAR_CON_ID	Varchar (15)	The parent row ID for the Contact table. The foreign key table is S_CONTACT.
PAR_ACCNT_ID	Varchar (15)	The parent row ID for the Account table. The foreign key table is S_ORG_EXT.
PAR_ORGGRP_ID	Varchar (15)	The row ID of the Household table. The foreign key table is S_ORG_GROUP.
PAR_ASSET_ID	Varchar (15)	The parent row ID of the FINCORP Account table. The foreign key table is S_ASSET.

Column Name	Data Type	Comments
EXT_SYS_ID	Varchar (15)	The external system ID. The foreign key table is S_CIF_EXT_SYS.
LAST_MOD_TS	UTC Date Time (7)	The last modified date and time.
DESC_TXT	Varchar (250)	Description.
CONFLICT_ID	Varchar (15)	Required. Siebel system field.
CREATED	UTC Date Time (7)	Required. Siebel system field.
CREATED_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
DB_LAST_UPD	UTC Date Time (7)	Required. Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Siebel system field.
LAST_UPD_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
MODIFICATION_NUM	Number (22)	Required. Siebel system field.

S_PARTY_REL_UPD Table

The following table shows details for the new S_PARTY_REL_UPD table which is used to configure survivorship attribute data for affiliation entity records.

Column Name	Data Type	Comments
ROW_ID	Varchar (15)	Required. Siebel system field.
AFFLTN_ID	Varchar (15)	Required. Entity row ID for Affiliation. The foreign key table is S_PARTY_REL.
FLD_VAL_RULE_ID	Varchar (15)	The field value row ID. The foreign key table is S_FLD_VAL_RULE.

Column Name	Data Type	Comments
EXT_SYS_ID	Varchar (15)	The external system ID. The foreign key table is S_CIF_EXT_SYS.
LAST_MOD_TS	UTC Date Time (7)	Last modified date and time.
DESC_TXT	Varchar (250)	Description.
CONFLICT_ID	Varchar (15)	Required. Siebel system field.
CREATED	UTC Date Time (7)	Required. Siebel system field.
CREATED_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
DB_LAST_UPD	UTC Date Time (7)	Required. Siebel system field.
DB_LAST_UPD_SRC	Varchar (50)	Siebel system field.
LAST_UPD	UTC Date Time (7)	Siebel system field.
LAST_UPD_BY	Varchar (15)	Required. Siebel system field. The foreign key table is S_USER.
MODIFICATION_NUM	Number (22)	Required. Siebel system field.

Modified Tables for Oracle Customer Hub (UCM) Version 8.2

The following tables were modified as part of Oracle Customer Hub (UCM) version 8.2 (for Siebel CRM 8.1.1.7 Quick Fix 07AN).

- *S_UCM_ASSET Table*
- *S_CIF_AST_MAP Table*
- *S_CONTACT_NAME Table*
- *S_CONTACT_IDNTY_DOC Table*
- *S_CONTACT_UCMX Table*
- *S_CONTACT Table*
- *S_DYN_HRCHY Table*
- *S_PARTY_UCMX Table*

- *S_UCM_ADDR_PER Table*
- *S_UCM_ATGP_RULE Table*
- *S_UCM_CONTACT Table*
- *S_UCM_CON_CHILD Table*
- *S_UCM_HRCHY Table*
- *S_UCM_HRCHY_REL Table*
- *S_UCM_ORG_EXT Table*
- *S_UCM_ORGGRP Table*
- *S_UCM_RULE_SET Table*

S_UCM_ASSET Table

The following table shows details of the S_UCM_ASSET table which is used for Financial Account Source Data and History (SDH).

Column Name	Data Type	Comments
ACCNT_CATEGORY_CD	Varchar (30)	The account category of a financial account. Example values include Asset, and Liability.
PAR_UCM_ACCNT_ID	Varchar (15)	Parent Oracle Customer Hub (UCM) Account ID. The foreign key table is S_UCM_ORG_EXT.
TRANSACTION_ID	Varchar	Oracle Customer Hub (UCM) asset for the S_ASSET table.

S_CIF_AST_MAP Table

The following table shows details of the S_CIF_AST_MAP table which is used for Financial Account cross references.

Column Name	Data Type	Comments
ACCNT_CATEGORY_CD	Varchar (30)	The account category of a financial account. Example values include Asset, and Liability.

S_CONTACT_NAME Table

The following table shows details of the S_CONTACT_NAME table.

Column Name	Data Type	Comments
CONTACT_ID	Varchar	Required. The foreign key table is S_CONTACT.
ROW_ID	Varchar	Required.
TYPE_CD	Varchar	Required. Values include Legal, Maiden, Diploma, and so on.
FST_NAME	Varchar	Person's first name.
LAST_NAME	Varchar	Person's last name.
MID_NAME	Varchar	Person's middle name.
SECOND_LAST_NAME	Varchar	Optional. Second last name. Used for contacts with two last names.
FULL_NAME	Varchar	Required. Concatenation of first, middle, last, and second last name.
ALIAS_NAME	Varchar	Optional. Alternate name used by person.
NICK_NAME	Varchar	Optional. Person's nickname.
PER_TITLE	Varchar	Personal Title. Values include Mr., Mrs., Dr., and so on.
PER_TITLE_SUFFIX	Varchar	Personal title suffix. Values include Esq., CPA, PhD, and so on.
PREF_FST_NAME	Varchar	Preferred first name.
HONORARY_TITLE	Varchar	Job title.
ROYAL_PREFIX	Varchar	Royal prefix. Rarely used, but it is required for royal contacts.
ROYAL_SUFFIX	Varchar	Royal suffix. Rarely used, but it is required for royal contacts.
EFF_START_DT	Date	Required. Effective start date. The default value is created date, but this value can be overwritten.
EFF_END_DT	Date	Effective end date.

Column Name	Data Type	Comments
CNTRY_FORMAT_MASK	Varchar	Country name format.

S_CONTACT_IDNTY_DOC Table

The following table shows details of the S_CONTACT_IDNTY_DOC table.

Column Name	Data Type	Comments
ROW_ID	Varchar	Required.
CONTACT_ID	Varchar	Required. Contact ID. The foreign key table is S_CONTACT.
ISS_COUNTRY_CD	Varchar	Country code.
ISS_STATE_CD	Varchar	State code.
TYPE_CD	Varchar	Required. National ID type, such as passport, driver's license, and so on.
IDENTITY_DOC_NUM	Varchar	Required. National ID. By default, this is not encrypted.
EFF_START_DT	Date	Required. Effective start date. The default value is created date, but this value can be overwritten.
EFF_END_DT	Date	Effective end date.

S_CONTACT_UCMX Table

The following table shows details of the S_CONTACT_UCMX table.

Column Name	Data Type	Comments
PR_NAME_ID	Varchar	Primary name ID. The foreign key table is S_CONTACT_NAME.
PR_IDENTITY_ID	Varchar	Primary Student ID. The foreign key table is S_CON_IDNTY_DOC.

Column Name	Data Type	Comments
CLASS_YEAR	Number	Graduation year. Current projected graduation year or latest graduation year for alumni.

S_CONTACT Table

The following table shows details of the S_CONTACT table.

Column Name	Data Type	Comments
MARITAL_STAT_DT	Date	Date the married status was changed.
PLACE_OF_DEATH	Varchar	Place of constituent's death.
DEATH_CERT_NUM	Varchar	Death certificate number.
VETERAN_FLG	Character	Required. Veteran benefits designation. Indicates whether a person is eligible for veteran's benefits or is a veteran.

S_DYN_HRCHY Table

The following table shows details of the S_DYN_HRCHY table.

Column Name	Data Type	Comments
LOCKED_FLG	Character	Dynamic hierarchy table.

S_PARTY_UCMX Table

The following table shows details of the S_PARTY_UCMX table.

Column Name	Data Type	Comments
RVW_STATUS_CD	Varchar	Party-privacy related attributes and other Oracle Customer Hub (UCM) attributes which apply to accounts, contacts and household entities.

S_UCM_ADDR_PER Table

The following table shows details of the S_UCM_ADDR_PER table.

Column Name	Data Type	Comments
REL_TYPE_CD	Varchar	Relationship between a contact and an address.
PAR_CON_ADDR_ID	Varchar	Foreign key for the best version address. The foreign key table is S_CON_ADDR.
TRANSACTION_ID	Varchar	Transaction ID.

S_UCM_ATGP_RULE Table

The following table shows details of the S_UCM_ATGP_RULE table.

Column Name	Data Type	Comments
RULE_MODULE_NAME	Varchar	Attribute group rule for Oracle Customer Hub (UCM) evaluation.
BUS_SERVICE_NAME	Varchar	Attribute group rule for Oracle Customer Hub (UCM) evaluation.
METHOD_NAME	Varchar	Attribute group rule for Oracle Customer Hub (UCM) evaluation.

S_UCM_CONTACT Table

The following table shows details of the S_UCM_CONTACT table.

Column Name	Data Type	Comments
MARITAL_STAT_DT	Date	Date marital status was changed.
DEATH_DT	Date	Date of constituent's death.
CLASS_YEAR	Number	Current projected graduation year or latest graduation year for alumni.

Column Name	Data Type	Comments
PLACE_OF_DEATH	Varchar	Place of constituent's death.
DEATH_CERT_NUM	Varchar	Death certificate number
VETERAN_FLG	Character	Required. Veteran benefits designation. Indicates whether a person is eligible for veteran's benefits or is a veteran.
ETHNICITY_CD	Varchar	Ethnicity code.
TRANSACTION_ID	Varchar	Transaction ID.
PR_COUNTRY	Varchar	Primary country of contact.
PRIM_STATE_CD	Varchar	Primary state of contact.
IMPORT_TASK_NUM	Varchar	Task number of the import operation.
IMPORT_TASK_ID	Varchar	Task ID of the import operation.
IMP_TASK_BATCH_NUM	Varchar	Batch number of the import operation.
VALID_FLG	Character	Indicates whether the record is valid.
IMP_BAT_UNIQUE_REF	Varchar	Unique reference number of batch import operation.
OPERATION_CD	Varchar	Operation code.
ERROR_CODE	Varchar	Error code.
ERROR_MSG_TEXT	Varchar	Text of any error messages generated.

S_UCM_CON_CHILD Table

The following table shows details of the S_UCM_CON_CHILD table.

Column Name	Data Type	Comments
PR_CHILD_FLG	Character	Indicates whether records are primary. The Primary flag is used to resolve the primary IDs in the S_

Column Name	Data Type	Comments
		CONTACT table after the records are propagated to the master records.

S_UCM_HRCHY Table

The following table shows details of the S_UCM_HRCHY table, which is a transaction staging table for Oracle Customer Hub (UCM) with integration of Oracle Hyperion Data Relationship Management.

Column Name	Data Type
UCM_UID	Varchar
NAME	Varchar
BV_LAST_UPD	UTC Date Time
BV_LAST_UPD_BY	Varchar
DYN_HRCHY_ID	Varchar
UCM_TYPE_CD	Varchar
DEFAULT_FLG	Character
UCM_TXN_DESC	Varchar
TRANSACTION_ID	Varchar
ERROR_CD	Varchar
ERROR_DESC	Varchar

S_UCM_HRCHY_REL Table

The following table shows details of the S_UCM_HRCHY_REL table, which is a transaction staging table for Oracle Customer Hub (UCM) integration with Oracle Hyperion Data Relationship Management.

Column Name	Data Type
PAR_ROW_ID	Varchar
UCM_UID	Varchar
PARTY_ID	Varchar
PAR_PARTY_ID	Varchar
UCM_TYPE_CD	Varchar
DYN_HRCHY_REL_ID	Varchar
UCM_TXN_DESC	Varchar
TRANSACTION_ID	Varchar
BV_LAST_UPD	UTC Date Time
BV_LAST_UPD_BY	Varchar
ERROR_CD	Varchar
ERROR_DESC	Varchar

S_UCM_ORG_EXT Table

The following table shows details of the S_UCM_ORG_EXT table.

Column Name	Data Type	Comments
PAR_PARTY_REL_ID	Varchar	Foreign key for the best version affiliation. Best version affiliation will be inserted in S_PARTY_PER instead of S_ORG_EXT.
AFFL_STATUS_DESC	Varchar	The status of an affiliation. The values include: Active, Inactive, or Error.
AFFL_STATUS_DESC	Varchar	A description that enhances the status information, for example, Inactive-withdrawal, inactive-terminated, and so on.

Column Name	Data Type	Comments
AFFL_RANK	Number	A five-digit number used by institutions to rank an affiliation.
REL_TYPE_CD	Varchar	Type of contact-account affiliation. The relationship between a contact and an account. The values can be: Student, Staff, Faculty, and Alumni.
REL_SUB_TYPE_CD	Varchar	Parent relationship code. It has similar values as those in the REL_TYPE_CD table. The hierarchy is maintained in the LOV table.
EFF_START_DT	Date	Effective start date. The default value is Created Date, but it can be overwritten.
EFF_END_DT	Date	Effective end date.
PRIMARY_FLG	Character	Required. Indicates whether the record is primary. When S_UCM_ORG_EXT is used to load child accounts of contacts, this indicator will be used to designate a primary account. The primary designation will be translated as the PR_OU_ID in S_CONTACT table or PR_AFFL_ID.
TRANSACTION_ID	Varchar	Transaction ID.

S_UCM_ORGGRP Table

The following table shows details of the S_UCM_ORGGRP table.

Column Name	Data Type	Comments
TRANSACTION_ID	Varchar	Stores the household record for the S_ORG_GROUP table.

S_UCM_RULE_SET Table

The following table shows details of the S_UCM_RULE_SET table.

Column Name	Data Type	Comments
PAR_OBJECT_CD	Varchar (30)	Parent entity object name.

Column Name	Data Type	Comments
END_DATE_FLD_NAME	Varchar (75)	Field name to pick expiry date or time from.

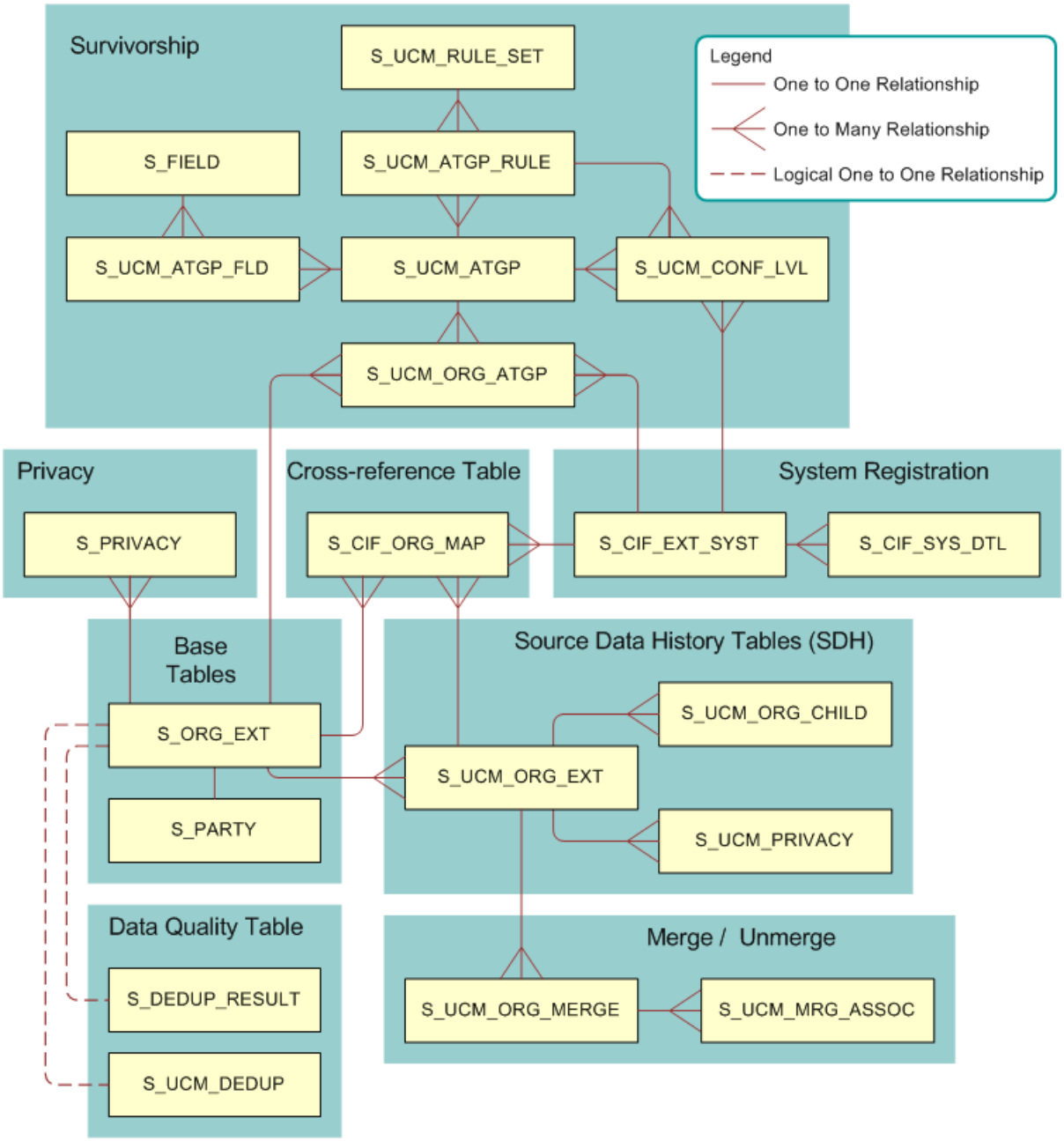
23 Oracle Customer Hub (UCM) Data Model Tables

Oracle Customer Hub (UCM) Data Model Tables

This chapter contains information about Oracle Customer Hub (UCM) data model tables. It includes the following topic: *About Oracle Customer Hub (UCM) Data Model Tables*.

About Oracle Customer Hub (UCM) Data Model Tables

The Oracle Customer Hub (UCM) data model defines how data used by Oracle Customer Hub (UCM) is stored in a database. The following figure displays the data model, which contains the following tables: Best Version, Source Data History (SDH), Survivorship, System Registration, Cross-reference, Privacy, Merge/Unmerge, Data Quality.



For more information on each table type in the Oracle Customer Hub (UCM) data model, see the following topics:

- *Best Version Tables* (base tables)
- *Source Data History (SDH) Tables*
- *Survivorship Tables*
- *System Registration Tables*
- *Cross-Reference Tables*
- *Privacy Tables*
- *Merge and Unmerge Tables*
- *Data Quality Tables*

Best Version Tables

Base tables store the mastered version of the customer record. Base tables have the following characteristics.

- Base table schemas are exact replicas of those in Siebel CRM.
- Cross-referencing and privacy data are stored as child records to the records in the base tables.
- Entities mastered in the base table can undergo merge and unmerge operations.
- Records undergo data-quality processes, such as cleansing, matching, and survivorship.
- Mastered records are continuously monitored by the batch-data quality processes to ensure high-quality data standards.

Source Data History (SDH) Tables

Source data history (SDH) tables hold incoming, best version, and historical data records. They have the following characteristics:

- Provide a mechanism to trace the life-cycle properties of records.
- Help maintain versioning of the base table records.
- Provide a staging area for data-quality processes before records move into base tables.
- Store the versioning for records going through the merge and unmerge process.

Note: External systems must integrate with these SDH tables. Oracle Customer Hub (UCM) integration revolves around integrating these SDH tables.

Survivorship Tables

Survivorship tables have the following characteristics:

- They contain customer data that originates in multiple systems.
  Different sources might contain conflicting data for the same attribute. For example, two subscribing systems might have the same customer name, but each might have a different address for the customer. Oracle Customer Hub (UCM) resolves these conflicts, ensuring the survival of the best-version record.
- The surviving record is that with the best available value for each attribute.
- Survivorship rules are an automated mechanism for consolidating customer data to produce a best-version record.
  Oracle Customer Hub (UCM) Survivorship Engine evaluates updates from each registered system, and applies rules to determine which value is best for each attribute.
- Survivorship rules can be created by a system integrator or a data steward.
  Oracle Customer Hub (UCM) allows two thresholds to be set depending on the scores from the matching process:
  - Records with high-match scores can be automerged by applying survivorship rules.

- Records with match scores in the middle will be sent to the data steward for review and resolution.
- Records with low-match scores will be left as separate records.

System Registration Tables

System registration tables have the following characteristics:

- Every operational application that connects to Oracle Customer Hub (UCM) must register with Oracle Customer Hub (UCM). After registration, a system's privileges and accessibility can be defined as: Query, Insert, Update, and Delete.
- The tables must define grant access to Oracle Customer Hub (UCM).

  Privileges are granted at the business object layer, in other words, a set of business objects are specified as accessible.

Cross-Reference Tables

Cross-referencing provides consistency by maintaining an external unique identifier for data from different systems to ensure universal uniqueness of records in the enterprise. Oracle Customer Hub (UCM) stores identification information about external systems that are registered to it.

Cross-reference tables have the following characteristics:

- A single Oracle Customer Hub (UCM) record maps to one or many records in any registered application.
- Oracle Customer Hub (UCM) supports cross-referencing for the following objects: Contact, Account, Household, and Financial Account, as well as account-address, account-financial accounts, contact-address, contact-financial accounts, contact-emails, contact-phones, and contact-names.
- Cross-reference entries are created automatically for the supported objects in SOAP and batch processes.
- External systems send requests to insert a record. If there is an existing cross-reference entry, then the entry is updated. If there is no existing cross-reference entry, then a new record is inserted. The system ID is the value assigned when the system was registered and system name is the associated name. There are multiple possible external IDs to account for situations where multiple records in an operational application map to a single record in Oracle Customer Hub (UCM). Only the first is stored in the ID tag in a message request, by default.

Privacy Tables

Privacy tables allow the enforcement of prebuilt rules to meet some U.S. regulations, such as Gramm-Leach Bliley Act, FCRA, FACTA, CA SB1. Privacy tables have the capability to capture historical versions of the privacy information for the master entities. Additionally, privacy tables have the following characteristics:

- Extended data model to capture communication preferences, track current and past privacy sharing statuses for various entities, such as affiliates, nonaffiliated third parties, brand, channels and telemarketing.
- The Privacy views are preconfigured with Contact, Account, Household, and Financial Account objects.
- Prebuilt workflows respond to customer life-cycle events and update privacy preferences proactively.

- Prebuilt, modular privacy framework and contents to create, maintain and deploy policies for each geography, federal, state or internal laws.
- Real-time or batch integration to receive and publish updated customer privacy sharing status to target systems, such as Siebel CRM, Marketing Campaigns, and so on.

Merge and Unmerge Tables

Merge and unmerge tables are populated during the merge process. Information stored in these tables is used for the Unmerge process. Additionally, merge and unmerge tables store the relationship information for the child entitles of the victim record during a merge.

Data Quality Tables

Data quality consists of data cleansing and data matching. Data cleansing standardizes data within Oracle Customer Hub (UCM). The typical functions are:

- Address validation and standardization.
- Capitalization.
- Abbreviations.
- Data matching identifies possible duplicate records for account, contact, and prospect records.
- Deduplication to describe matching in user properties and system parameters.

24 Additional License Restrictions

Additional License Restrictions

This chapter describes the license restrictions that apply to Oracle Master Data Management Applications. These restrictions supersede all other grants and restrictions. This chapter includes the following topic:

- *Additional License Grants and Restrictions Regarding Oracle Master Data Management Applications*

Additional License Grants and Restrictions Regarding Oracle Master Data Management Applications

Oracle Master Data Management 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 Oracle Master Data Management 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 Oracle Master Data Management Applications solely in accordance with the Documentation to configure only those Business Objects and Business Components (BusComps) included with the licensed Oracle Master Data Management 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 Oracle Master Data Management Applications, solely through Object Manager and EAI, EIM, and Business Process Designer.
- Notwithstanding anything to the contrary, Customer shall have no right to (i) extend or modify the Siebel Data Model other than through Siebel Tools; (ii) use the tables in the Siebel Data Model for any Siebel Programs not licensed by Customer (for example, Siebel Partner Relationship Manager); (iii) use the Siebel Tools for general application development purposes, or (iv) use or access Oracle Master Data Management Applications (including the Siebel Data Model) for non-Siebel CRM (Customer Relationship Management) applications.
- Customer will not use this Oracle Master Data Management Applications licensing model to avoid paying Program License Fees for (1) full use of the functionality contained in other Siebel programs other than Oracle Master Data Management Applications, or (2) the ability to use Oracle Master Data Management Applications Programs outside of the specific license grants as listed.
- Oracle Customer Hub (UCM) licensing provides unlimited read-only user access. If you require UI access to create or update data, then you must license the Data Steward product.

The following table sets forth the Business Components included in Oracle Master Data Applications.

Master Data Module	Business Components
Oracle Customer Hub (UCM)	Party
	Party Contact
	Party Relationship To
	Account
	Contact
	Contact Relationship
	Household
	CUT Address
	Personal Address
	Channel Partner
	Partnership Contact
	Opportunity
	Position
	Employee
	Expense
	Communications Address
	Alternate Phone
Oracle Activity Hub	Action
	Contact Note
	Contact Private Note

Master Data Module	Business Components
	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

Master Data Module	Business Components
	FINCORP Client Contact Management
	FIN Contact Income
	FIN Contact Employment
	FINS Financial Statement Contact
	FINS Financial Accounts for Contacts
	FINCORP Account
	FINCORP Account Contact
Oracle Activity Hub (continued)	FINCORP Contact Education
	FINCORP Contact Experience
	UCM FINCORP Account Source Data and History
	UCM FINCORP Account Contact Source Data History
	UCM FINCORP Account Contact Garage Address Source Data and History
	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

Master Data Module	Business Components
	FINS Billing Accounts
	FINS Billing Accounts (No Contact Copy)
	FINS Contact Household Contact
	FINS Household Billing Accounts (Read Only)
	Comm Invoice Profile
Oracle Activity Hub (Hospitality Business Components)	Accounts: Accounts
	Accounts: TNT SHM Account Travel Profile
	Contacts: Contact (B2B Contacts)
	Contacts: TNT SHM Contact Travel Profile
	Property: SHM
	Function Spaces: Internal Product
	Functions: TNT SHM Function
	Cluster: TNT SHM Cluster Employee
	Cluster: TNT SHM Cluster
	Function Space Inventory: TNT SHM FSI Bookings
	Function Space Inventory: TNT SHM FSI Auth Lvl
Oracle Sales Hub	Complex Product
	Internal Product
	Internal Product Attributes
	Internal Product Attachment
	Internal Product Image

Master Data Module	Business Components
	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

Master Data Module	Business Components
	Opportunity
	Opportunity (Orders)
	Competitor
	Competitor 2
	Quote
Oracle Sales Hub (continued)	Quote Attachment
	Quote Item
	Key Feature
	Admin Product Comparison
	Sales Tool by Product
	XA Attribute Value
	Cost List Item
	Product Defect
	Partner Product
	Revenue
	Orders
	Order Entry (Orders)
	Order Entry (Line Items)
	Order Entry (Order Terms)
	Decision Issue
	Decision Issue Attachment

Master Data Module	Business Components
	Opportunity Forecast
	Opportunity Forecast Detail (Wide)
	Opportunity Forecast Line
	Opportunity Person Forecast
	Opportunity Product Forecast
Oracle Service Hub	Service Request
	Related Service Request
	Solution
	Related Solution
	Resolution Activity
	SR Resolution Item
	RMA
	Service Agreement Account
	Service Agreement
	Service Agreement Contact
Oracle Service Hub (continued)	Service Agreement Product
	Service Agreement Attachment
	Product Defect
	Decision Issue
	Customer Survey
	Customer Product

Master Data Module	Business Components
	Activity Plan
	Related Service Request
	FIN Service Request
Oracle Marketing Hub	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
Oracle Marketing Hub (continued)	Prospect Substitution
	List Management
	List Mgmt List Member Prospect
	List Mgmt Prospective Contact
	List Mgmt Prospective Contact Mkt Segment
	List Mgmt Prospective Contact (Attributes)

Master Data Module	Business Components
	LOY Attribute Definition
	LOY Household
	LOY Loan Repayment
	LOY Member
	LOY Member Tier
	LOY Membership Card
	LOY Point Block
	LOY Product Points
	LOY Program
	LOY Promotion
	LOY Promotion Bucket
	LOY Statement
	LOY Tier
	LOY Tier Class
	LOY Transaction
	LOY Transaction Accrual Item
	LOY Transaction Redemption Item
	LOY Voucher
	Offer
	Response
	Segments

Master Data Module	Business Components
	Programs
	Program Container
	Program Container Occurrences
	Program Occurrences
Oracle Field Service Hub	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

Master Data Module	Business Components
	Scheduling
Oracle Privacy Hub	UCM FINCORP Account Privacy
	UCM FINCORP Account Privacy Source Data and History
	UCM FINCORP Account Contact Privacy
	UCM FINCORP Account Contact Privacy Source Data and History
	UCM FINCORP Account Contact Address
	UCM Account Privacy
	UCM Account Privacy Source Data and History
	UCM Contact Privacy
	UCM Contact Privacy Source Data and History
Oracle Higher Education Constituent Hub	UCM HE Constituent Name
	UCM HE Constituent Identification
	UCM HE Constituent Address
	UCM HE Constituent Affiliation
	UCM HE Constituent Name Source Data and History
	UCM HE Constituent Identification Source Data and History
	UCM HE Constituent Address Source Data and History
	UCM HE Constituent Affiliation Source Data and History
	UCM Alternate Phone Source Data and History
	UCM Communication Address Source Data and History
	Contact (for Constituent)

Master Data Module	Business Components
	UCM Contact Source Data and History (for Constituent SDH)
	UCM Account Source Data and History (Added Search Specification to differentiate between Account SDH and Affiliation SDH which are configured on the same SDH table.)
	Alternate Phone (for Constituent Phone)
	Communication Address (for Constituent Email)

25 Troubleshooting Oracle Master Data Management Applications

Troubleshooting Oracle Master Data Management Applications

This chapter describes how to troubleshoot Oracle Master Data Management Applications. It includes the following topic:

- *Troubleshooting Oracle Master Data Management Applications*

Troubleshooting Oracle Master Data Management Applications

To resolve the problem, look for it in the list of error messages in the following table.

Error Number	Error Message	Diagnostic Step or Cause	Solution
SBL-IAI-00427	%1 Checking if privilege is allowed for %2 operation	Checking if a system has the privilege to perform the operation.	An informational message. No action is required.
SBL-IAI-00428	%1 Checking if security is enforced	Checking if security is enabled in Oracle Customer Hub (UCM).	An informational message. No action is required.
SBL-IAI-00429	%1 Checking system registration for system %2	Checking if a system is registered in Oracle Customer Hub (UCM).	An informational message. No action is required.
SBL-IAI-00430	%1 Generating Unique Id for %2	Oracle Customer Hub (UCM) is generating a new, unique ID for this message.	An informational message. No action is required.
SBL-IAI-00431	%1 Inserting reference record for %2 with Id %3	Inserting external ID record.	An informational message. No action is required.
SBL-IAI-00432	%1 Failed to find %2	A business service failed to find an object.	Make sure the object exists.

Error Number	Error Message	Diagnostic Step or Cause	Solution
SBL-IAI-00433	%1 Failed to find integration object instance	A business service failed to find an integration object instance. This error indicates that no valid data was received.	Check that an integration object instance exists.
SBL-IAI-00434	%1 Failed to find Integration Object name	A business service failed to find an integration object name.	Check that an integration object instance exists in the property set.
SBL-IAI-00435	%1 Failed to find message instance under XML Hierarchy	Not able to find a message in the input XML.	Check that the XML document sent to Oracle Customer Hub (UCM) contains a message body.
SBL-IAI-00436	%1 Failed to find System Id in input message	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.	Verify that the input property set contains a child XML hierarchy property set.
SBL-IAI-00438	%1 Failed to find Privilege record for %2 object in UCM Administration	Required privilege is not allowed on the object.	Check in the Oracle Customer Hub (UCM) Administration views that the system is registered and has privileges defined.
SBL-IAI-00439	%1 Failed to find %2 in System Registration	Unable to find the system in the 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 system ID is registered in the Oracle Customer Hub (UCM) Administration views.
SBL-IAI-00440	%1 Updating Cross-Reference record for %2 with Id %3	Updating the external ID for the record.	An informational message. No action is required.
SBL-IAI-00442	%1 Found match	Match found.	An informational message. No action is required.
SBL-IAI-00443	%1 Found no match	No match found.	An informational message. No action is required.
SBL-IAI-00444	%1 Found multiple instances within the input	Multiple accounts or contacts found in the input	Check the input data and verify that only one

Error Number	Error Message	Diagnostic Step or Cause	Solution
		from SDH record. Currently Oracle Customer Data Management supports only one record.	instance, for example, an account or contact, is supplied.
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-00446	%1 has no Manual Threshold definition in user property	No manual threshold definition found in the user property of Oracle Customer Data Managment service.	Check if a manual threshold definition exists in Data Quality Manager business service user properties, and make sure the value is between 0 and 100.
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 that it has been configured correctly.
SBL-IAI-00448	%1 PropertySet doesn't match the type %2	The type for the property set is not correct. Data quality match must return a property set of type, Match Info.	Check the output property set of your deduplication business service. The property set must have a type of Match Info.
SBL-IAI-00449	%1 Search failed to return results from %2	The search failed to return an active row and one active row is expected.	You might receive this error in the following cases: • When updating the Duplicate flag in Oracle Customer Hub (UCM) table, it does a query for the duplicate record and does not find it. • When querying for the cleansed version of the source data, Oracle Customer Hub (UCM) does a query for the cleansed record and does not find it. • When querying the business component within the Deduplication business object, Oracle Customer Hub (UCM) queries for the field used in deduplication and does not find the field. • When cleaning the selected data

Error Number	Error Message	Diagnostic Step or Cause	Solution
			in the business component within the deduplication business object, Oracle Customer Hub (UCM) does a query for the business component record and does not find a record. • When clearing the Oracle Customer Hub (UCM) source table Duplicate flag, Oracle Customer Hub (UCM) does a query for the duplicate record and does not find it.
SBL-IAI-00451	%1 Failed to find %2 with %3	A business service failed to find the object.	Make sure an object exists for the ID supplied.
SBL-IAI-00455	%1 Failed to find a valid Rule Set in UCM Administration	A business service failed to find a configured default rule set.	Make sure a default and nonexpired rule set for the object is configured. For more information, see *Configuring the Survivorship Settings for Oracle Customer Hub (UCM)*.
SBL-IAI-00457	%1 Failed to find Default Attribute Group	A business service failed to find a configured default attribute group.	Make sure a default and nonexpired attribute group for the object is configured. For more information, see *Configuring the Survivorship Settings for Oracle Customer Hub (UCM)*..
SBL-IAI-00458	%1 Invalid value %2 for parameter Batch Size	A business service found an invalid value for the input parameter, Batch Size.	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 business service found an invalid value for the input parameter, Sleep Time.	Make sure the input argument, Sleep Time, is an integer and equal to or greater than zero.
SBL-IAI-00460	%1 Failed to find required parameter Object Type	A business service failed to find required input parameter, Object Type.	Make sure the input argument has an object type. The object types are: Contact, Account, or Household.

Error Number	Error Message	Diagnostic Step or Cause	Solution
SBL-IAI-00461	%1 Invalid Match Id %2 for AutoMatch	A business service found an invalid match ID for AutoMatch.	Make sure there is a valid and not null Match ID value supplied for the AutoMatch function.
SBL-IAI-00462	%1 Failed to find current record for incoming record with SDH Id %2	Unable to query Best Version record from base table while preparing the history record. This might be because no record is satisfying the user key in IO. For example, similar fields are not present in the input, or fields are empty in input).	Make sure the incoming record contains either a Party_UID or Row_Id or both. If any of these ID values are supplied, then check that there is only one matching record in the SDH tables with that value.
SBL-IAI-00463	%1 Failed to link source record with Id %2 to base record	When linking the SDH record to the best version record, unable to update Best Version ID in the SDH record in the SDH table.It might be because no user key fields are in the input.	Make sure the Oracle Customer Hub (UCM) UID is valid.
SBL-IAI-00465	%1 Found no User Property key definition for %2 in Dispatch Map	Cannot find the user property key in the integration object definition of the dispatch map.	Check that there is a user property defined in the Oracle Customer Hub (UCM) Dispatch Map integration object, such as CIFDispMap.
SBL-IAI-00466	%1 Found no Integration Object definition for %2	No integration object definition was found in the repository. Check that you have the correct integration object name in the message.	Make sure the integration object name is correct.
SBL-IAI-00467	%1 Found multiple selections from UI	Multiple selection from the deduplication UI.	Select only one row in the applet when running the Link and Update function.
SBL-IAI-00474	%1: Child object %2 with Id %3 not found. Record not reparented	Reparenting one of the child records failed because the child record ID was not found. The selected record could not be unmerged.	An informational message. No action is required.
SBL-IAI-00475	%1 Failed to reparent child %2	Reparenting one of the child records failed because the child record ID was not found.	An informational message. No action is required.

Error Number	Error Message	Diagnostic Step or Cause	Solution
SBL-IAI-00478	Failed to find Case Type	The valid case type values are: NoMatch, AutoMatch, and Review.	An informational message. No action is required.
SBL-IAI-00479	%1: Unmerge of parent record failed because merged SDH victim record is not found for object %2	Unmerging the parent record failed.	An informational message. No action is required.
SBL-IAI-00480	%1: Unmerge of parent record failed because surviving record is not refreshed for object %2	When an unmerge is done after a merge operation and without an update operation, then the surviving record is refreshed. If the refresh fails, then this error occurs.	An informational message. No action is required.
SBL-IAI-00481	Parent object %1 with Id %2 not found	The unmerge operation failed. The parent record was not found. This error might have occurred because the parent record was deleted.	An informational message. No action is required.

The following table is a continuation of the list of Oracle Master Data Application error messages.

Error Number	Error Message	Diagnostic Step or Cause	Solution
SBL-IAI-00482	Unmerge complete: Deletion of last unmerged children for SDH Id %1 from merge table failed for Object %2	The unmerge process completed successfully, but the children records were not deleted. Deleting the child records from the merge table has failed. It is most likely that you have to resolve a database problem.	Review the Oracle Customer Hub (UCM) log files for messages on why the child records were not deleted. You must work with your database administrator to manually delete the child records from the database. The SQL statement is in the following format: `DELETE FROM <tbo>.S_ UCM_XXX_MERGE WHERE UCM_XXX_ID=<ID value from message>` Do the following: • If you are working with accounts, then substitute *XXX* with *ORG*. • If you are working with *contacts*, then

Error Number	Error Message	Diagnostic Step or Cause	Solution
			substitute *XXX* with *CON*.

SBL-IAI-00483	Unmerge complete: Updating unmerged record to Unmerge state failed for SDH Id %1 for Object %2	The unmerge process completed successfully, but the victim record status was not changed from Merged to Unmerged. It is most likely that you have to resolve a database problem.	Review the application log files for detailed messages on why the status was not updated. You must work with your database administrator to manually update the status value in the database.
			
			If you are working with accounts, then the SQL statement is as follows:
			
			`UPDATE <tbo>.S_UCM_ CONTACT SET UCM_TYPE_CD='Unmerged' WHERE ROW_ID=<ID value from message>`
			
			If you are working with contacts, then then SQL statement is as follows:
			
			`UPDATE <tbo>.S_UCM_ ORG_EXT SET UCM_TYPE_CD='Unmerged' WHERE ROW_ID=<ID value from message>`

SBL-IAI-00484	Message is received and queued for batch processing	When a message is received by Oracle Customer Hub (UCM) in batch mode, the message is queued for batch processing.	An informational message. No action is required.
SBL-IAI-00485	valid protocol type	Passed as parameter 2 to error message: IDS_SIA_ CIF_MISSING_GENERAL.	An informational message. No action is required.
SBL-IAI-00486	%1 Creating Default Attribute Group Data	Creating a default attribute group data for the new object instance.	An informational message. No action is required.
SBL-IAI-00487	%1 Getting Attribute Group Data	Retrieving the attribute group data for the object.	An informational message. No action is required.
SBL-IAI-00488	%1 Getting survivorship rule	Retrieving the active or default survivorship rule.	An informational message. No action is required.

Error Number	Error Message	Diagnostic Step or Cause	Solution
SBL-IAI-00489	%1 Inserting Attribute Group Data	Inserting attribute group data for the corresponding object.	An informational message. No action is required.
SBL-IAI-00490	%1 Processing base fields	Processing and merging the supported base fields in the object.	An informational message. No action is required.
SBL-IAI-00495	Prepare Rule Engine Input	A business service Method Name. The first step is to prepare an input argument for the rules engine.	Check if the argument is missing for this method.
SBL-IAI-00496	Siebel Message	A business service Method Argument.	Provide this argument for the business service method.
SBL-IAI-00497	Object Id	A business service Method Argument.	Provide this argument for the business service method.
SBL-IAI-00498	Top Level Business Component Name	A business service Method Argument.	Provide this argument for the business service method.
SBL-IAI-00499	Invalid input argument %1 found in the %2 method in Business Service %3	The input argument is invalid.	Check that the input argument is not empty and contains valid data.
SBL-IAI-00500	Prepare History Record	A business service method name. The method used to prepare the history record.	Check if the argument is missing for this method.
SBL-IAI-00501	Failed to validate Account Hierarchy relationship - can't find %1 with parent %2 in the relationship	The current global account hierarchy has an account that does not exist in the hierarchy. For example, if you got the error message: Failed to validate Account Hierarchy relationship - can't find B with parent D in the relationship. Given the following relationships for a hierarchy H (Accnt, Parent Accnt): (A, B) and (B, D), the relationship (B, D) is invalid because Account D is not in the relationship. Account D might not exist.	Verify that the relationship between the accounts identified in the error message is valid and that both accounts exist before copying the account hierarchy.

Error Number	Error Message	Diagnostic Step or Cause	Solution
SBL-IAI-00502	%1 service: Existing %2 Best Version Record with RowId = %3 not found for System Id %4	Oracle Customer Hub (UCM) failed to find an existing record for an incoming message.	Check if the external ID value in the incoming message has a correct cross-reference ID and that this ID exists in Oracle Customer Hub (UCM). This error might occur if the best version record is deleted and the cross-reference record still exists in the database.
SBL-IAI-00503	%1 service: Delete operation from the System Id %2 cannot be completed because record with Row Id='%3' is not found	A delete operation from an external system failed because the record does not exist in Oracle Customer Hub (UCM).	Make sure that at least one user key field is present in the input message and the corresponding record exists in Oracle Customer Hub (UCM).
SBL-IAI-00504	%1 service cannot interpret the output from %2 service	A business service cannot interpret the input.	Make sure that the business service is configured correctly and that it returns a valid output argument.
SBL-IAI-00505	Service %1 method %2 requires an input parameter %3 which is missing	Invalid input parameter.	Make sure that input parameter has a valid value and is not empty.
SBL-IAI-00506	%1 Failed to find MatchOutput in MatchAllOutputs PropertySet in parameter XMLHierarchy in method %2	A business service method failed to find the MatchOutput property set in the MatchAllOutputs property set in the XMLHierarchy property set.	Make sure that the MatchAllOutputs child property set in input argument XMLHierarchy has one or more MatchOutput child property sets. Check that the output from the matching service is valid. Check your log file for more information.
SBL-IAI-00507	%1 service: Matching Component user property is not defined	A business service was not configured correctly for the Match method.	Make sure the corresponding user properties, Matching Component and Matching Field, are defined in Data Quality Manager business service and it has a valid value.
SBL-IAI-00508	%1 service: Exact Match Object user property is not defined	A business service was not configured correctly for the ExactMatch method.	Make sure the user property, ExactMatch Object, is defined in Data Quality Manager business service and it has a valid value.

Error Number	Error Message	Diagnostic Step or Cause	Solution
SBL-IAI-00509	Parameter Case Type is required for method %2 in service %1	A business service method failed to find required input parameter called, Case Type.	Make sure the input argument has a case type parameter defined. The case types are: Default or NoMatch, AutoMatch or Review.
SBL-IAI-00510	%1 Invalid value %2 for input parameter Position	A business service found an invalid value for the input parameter, Position.	Make sure the input argument, Position, is an integer, not empty, equal to or greater than zero, and less than the number of records in the input property set.
SBL-IAI-00511	%1 Failed to find MatchAllOutputs in input parameter XMLHierarchy in method %2	A business service method failed to find the MatchAllOutputs property set in the input XMLHierarchy property set.	Make sure the input argument XMLHierarchy has at least one MatchAllOutputs child property set. Check that the output from the matching service is valid. Check your log file for more information.

The following table is a continuation of the Oracle Master Data Application list of error messages.

Error Number	Error Message	Diagnostic Step or Cause	Solution
SBL-IAI-00512	%1 service failed to process the record with SDH Row Id=%2	A business service failed to process the record. The SDH ID is the row ID of the Incomplete status record in the History version.	Go to the Oracle Customer Hub (UCM) Administration screen, Transaction History View, and search for records with an incomplete status. Find the row ID of those records, and check the log file for the error message to determine the cause of the error.
SBL-IAI-00513	%1 service: Internal error when %2 method is invoked	Fatal error encountered when the business service method is invoked.	For a batch process, restarting Oracle Customer Hub (UCM) Publish Subscribe task might solve the problem. For a real-time process, you must manually reprocess it because the message might have been lost.
SBL-IAI-00514	%1 service: Failed to find the PartyPackage tag in incoming message	The XML input message received does not contain the PartyPackage tag.	Make sure that input XML message has a PartyPackage tag defined in it.

Error Number	Error Message	Diagnostic Step or Cause	Solution
SBL-IAI-00515	%2 is not a registered object type with %1 service	The external name of the integration object name does not match the registered object types.	Add a new opbject type to CIF_BUSINESS_OBJECT_ TYPE LOV. **Note:** Make sure that the external name of the integration object definition matches the LOV values.
SBL-IAI-00516	%1 service: Cannot find JMS secure queue authentication parameters for queue: %2	Cannot find the user names or passwords for JMS secure queue connection.	JMS secure queue user names and passwords can be provided in the configuration file.
SBL-IAI-00517	%1: '%2' System does not have privilege for %3 operation	The system does not have the privilege for the specified operation.	Navigate to the System Registrations screen, and define the privileges for this system.
SBL-IAI-00518	%1 Failed to find required parameter Search Spec	A business service failed to find the required input parameter search specification.	Make sure the input argument does not have an empty search specification.
IDS_SIA_UCM_ TM_MISSING_ EXT_ SYST_ID	%1 Failed to find parameter External System Id.	A business service failed to find the input parameter external system ID.	Make sure the input argument does not have an empty external system ID.
DS_SIA_UCM_ TM_MISSING_ OPERATION	%1 Failed to find parameter Operation.	A business service failed to find the input parameter, Operation.	Make sure the input argument does not have an empty operation.
IDS_SIA_UCM_ TM_MISSING_ OP_SVC_ METH	%1 Failed to find Service/ Method Names for %2 Operation.	A business service failed to find Service or Method Names for the specified operation in the business service user properties.	Make sure the business service user property exists and has a valid value.
IDS_SIA_UCM_ TM_INVALID_FORMAT_ OP_SVC_ METH_ARG	%1 Found Service Method Argument value of %2 Operation not ending with ;	A business service found an invalid format (not ending in a semicolon [;]) of the Service Method Argument in the business service user property for the specified operation.	Make sure the business service User property value has a valid format.
IDS_SIA_UCM_ MUM_PUBLISH_ FAILED	%1 :Failed to Publish UCM Merge operation between records with Row Ids %2 and %3.	Publishing the merge operation (for the data steward) failed in Oracle Customer Hub (UCM). It might be because of	Check if there is a problem with the message or publish subscribe service.

Error Number	Error Message	Diagnostic Step or Cause	Solution
		an internal error in the message structure or the publish subscribe service.	
IDS_UCM_ MISSING_ REQUESTER_ID	%1 Failed to find Business Service User Property RequesterSystemId for SOAP input.	A business service failed to find the user property, RequesterSystemId, for the SOAP input.	An informational message. No action is required.
IDS_UCM_ MESSAGE_TYPE	%1 Failed to find input Message Type.	A business service failed to determine the input message type. Without a valid message type, Publish/Subscribe method fails.	Make sure the input is a valid Simple Object Access Protocol (SOAP) message.
IDS_UCM_ERR_ MISSING_ MIDDLEWARE_INFO	%1 Failed to find required Transport parameters for Middleware System.	A business service failed to find the required input transport parameters for the middleware system. It cannot find the proper connect information for the middleware system. Without the connect parameters to the middleware system, the Publish and Subscribe functionality fails.	Make sure the transport type and the connect information are provided for the middleware system in the Systems Registration-CIF Administration screen.
IDS_UCM_ XR_MISSING_ XREF_RECS	%1 Failed to find cross-reference records in input instance.	A business service failed to find cross-reference records in the input integration object instance.	Make sure at least one cross-reference record is present in the input message.
IDS_UCM_XR_ PARENT_ID_ NOT_ QUERIED	%1 Failed to find Id in queried parent record.	A business service failed to find the ID field in the parent record being queried.	Make sure the ID field is configured for the root component in the integration object definition.
IDS_UCM_XR_ QUERY_PARENT_ FAILED	%1 Failed to query the parent record.	A business service found zero or multiple records matching the parent record in the input integration object instance.	Make sure valid parent user key fields are present in the input message and the corresponding record exists in Oracle Customer Hub (UCM).
IDS_UCM_XR_ QUERY_XREF_ REC_ FAILED	%1 Operation failed for cross-reference record which is not found.	A business service failed to complete the operation because the cross-reference record does not exist in Oracle Customer Hub (UCM).	Make sure valid cross-reference fields are present in the input message and the corresponding record exists in Oracle Customer Hub (UCM).

Error Number	Error Message	Diagnostic Step or Cause	Solution
IDS_UCM_ERR_ TOO_MANY_ MIDDLEWARE	%1 :Validated the list of registered systems. Too many Middleware Systems found.	Validating the list of registered systems first.	Make sure that there is only one middleware system.
IDS_UCM_ERR_ COMMIT_ MIDDLEWARE	%1 :Cannot commit record as a Middleware System is already found.	More than one middleware system is not supported.	Undo the current record because more than one middleware system cannot be supported.
IDS_UCM_ XMIDDLEWARE_ PROTOCOL	%1 :Non-Middleware Systems cannot use PROXY protocol transport.	Nonmiddleware systems cannot use the proxy protocol transport.	Undo the protocol type.
IDS_UCM_ XMIDDLEWARE_ MSG	%1 :Middleware System found.Changing the message type of all systems to SOAP.	When a middleware system is present all nonmiddleware systems have the SOAP message type.	Change the message type of all the nonmiddleware systems to the SOAP message type.