# Siebel

## Data Quality for Oracle Customer Hub Guide

July 2025

Siebel
Data Quality for Oracle Customer Hub Guide

July 2025

Part Number: F87451-02

Copyright © 1994, 2025, Oracle and/or its affiliates.

Authors: Siebel Information Development Team

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or “commercial computer software documentation” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government’s use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

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

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

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

# Contents

**Preface** i

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

What's New in Siebel Data Quality Guide for Oracle Customer Hub, Siebel CRM 25.7 Update 1
What's New in Siebel Data Quality Guide for Oracle Customer Hub, Siebel CRM 21.4 Update 1
What's New in Siebel Data Quality Guide for Oracle Customer Hub, Siebel CRM 20.1 Update 1
What's New in Siebel Data Quality Guide for Oracle Customer Hub, Siebel CRM 19.1 Update 2

**2 Overview of Data Quality** 3

Overview of Data Quality 3
Data Profiling 3
Data Parsing and Standardization 4
Data Matching and Data Cleansing 4
Data Quality Products for Data Matching and Data Cleansing 5
How Data Quality Relates to Other Entities in Siebel CRM 9

**3 Data Quality Concepts** 11

Data Quality Concepts 11
Data Cleansing 11
Data Matching 12
Match Key Generation 13
Identification of Candidate Records 15
Calculation of Match Scores 16
Displaying Duplicates 16
Fuzzy Query 17

**4 Enabling and Disabling Data Matching and Data Cleansing** 19

Enabling and Disabling Data Matching and Data Cleansing 19
Levels of Enabling and Disabling Data Cleansing and Data Matching 19
Enabling Data Quality at the Enterprise Level 21

Specifying Data Quality Settings 22
Enabling Data Quality at the Object Manager Level 25
Enabling Data Quality at the User Level 27
Disabling Data Cleansing for Specific Records 28
Enabling and Disabling Fuzzy Query 28
Identifying Mandatory Fields for Fuzzy Query 30

5 Configuring Data Quality with Universal Connector 31
Configuring Data Quality with Universal Connector 31
Data Quality Configuration Overview 31
Installing the Universal Connector 32
Process of Configuring New Data Quality Connectors 34
Configuring Vendor Parameters 37
Mapping of Vendor Fields to Business Component Fields 37
Configuring the Windows Displayed in Real-Time Data Matching 40
Configuring the Mandatory Fields for Fuzzy Query 41
Data Quality User Properties 42
Optimizing Data Cleansing Performance 49
Optimizing Data Matching Performance 50

6 Configuring Data Quality with Oracle Enterprise Data Quality 53
Configuring Data Quality with Oracle Enterprise Data Quality 53
Overview of Oracle Enterprise Data Quality 53
Installing Oracle Enterprise Data Quality (EDQ) 54
Process of Configuring Oracle Enterprise Data Quality for Matching and Address Verification 55
Process of Configuring the Staging Database for Oracle Enterprise Data Quality 59
Process of Configuring Oracle Enterprise Data Quality to Find Contact and Address Matches 60
Vendor Parameters for Oracle Enterprise Data Quality 61
Oracle Enterprise Data Quality Field Mappings for Siebel Business Components 63
Configuring Third-Party Key Generation for Oracle Enterprise Data Quality 72
Enabling Support of Multiple Child Entity Deduplication for Oracle Enterprise Data Quality 73
Data Format for Deduplication of Multiple Child Entities 73
Overview of Configuring Oracle Enterprise Data Quality for Deduplication of Child Entities 76
Process of Configuring Additional Child Entities for Deduplication 77
Sample Message from Siebel CRM to EDQ Siebel Connector for Additional Child Entity Deduplication 82
Overview of Configuring Social Media Child Objects for Matching 83
Sample Message from Siebel CRM to EDQ Siebel Connector for Social Media Child Object Matching 83

## 7 Configuring Data Quality with Oracle Data Quality Servers 85

Configuring Data Quality with Oracle Data Quality Servers 85
Process of Installing the Oracle Data Quality Matching Server 86
Upgrading to Informatica Identity Resolution 9.01 103
Configuring Oracle Data Quality Matching Server 105
Universal Connector Parameter and Field Mapping Values for Oracle Data Quality Matching Server 106
Configuring a New Field for Real-Time Data Matching 110
Loading Data Incrementally 113
Process of Configuring Data Synchronization Between Siebel CRM and Oracle Data Quality Matching Server 115
Siebel CRM DQ Sync Action Sets 117
Configuring Deduplication Against Multiple Addresses 137
Configuring Multiple Language Support for Data Matching 139
Configuring Multiple Mode Support for Data Matching 142
Sample Configuration Files for Oracle Data Quality Matching Server 144
Process of Installing the Oracle Data Quality Address Validation Server 147
Configuring Siebel CRM for the Oracle Data Quality Address Validation Server 152
Universal Connector Parameter and Field Mapping Values for Oracle Data Quality Address Validation Server 154
Sample Configuration File for Oracle Data Quality Address Validation Server 157
Siebel CRM ISSSYNC Action Sets 158
Activating Action Sets in Your Siebel Application 184
Sample SQL Scripts 185
Sample SiebelDQ.sdf File 194

## 8 Administering Data Quality 201

Administering Data Quality 201
Data Quality Modes of Operation 201
Data Quality Rules 203
Data Quality Batch Job Parameters 205
Cleansing Data Using Batch Jobs 207
Matching Data Using Batch Jobs 208
Merge Algorithm in the Object Manager Layer 211
Merging Duplicate Records 213
Process of Merging Duplicate Records 214
Using Fuzzy Query 216
Calling Data Matching and Data Cleansing from Scripts or Workflows 218
Troubleshooting Data Quality 223
Viewing Parameter and Field Mapping Values for Universal Connector 224

9 Related Data Quality Objects in Siebel Tools 227
Related Data Quality Objects in Siebel Tools 227
Siebel Data Quality Objects 227

10 Universal Connector API 229
Universal Connector API 229
About the Terminology in This Chapter 229
About the Universal Connector API 230
About Universal Connector API Terminology 230
Vendor Libraries 230
Connector Initialization and Termination Functions 231
Session Initialization and Termination Functions 232
Parameter Setting Functions 233
External Key Generation Function 235
Error Message Function 238
Real-Time Data Matching Functions 238
Batch Mode Data Matching Functions 241
Real-Time Data Cleansing Function 245
Batch Mode Data Cleansing Function 246
Data Matching and Data Cleansing Algorithms 246

11 Finding and Using Data Quality Information 249
Finding and Using Data Quality Information 249
Data Quality Resources 249
Data Quality Seed Data 250

# 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 Siebel Data Quality Guide for Oracle Customer Hub, Siebel CRM 25.7 Update

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

***What's New in Siebel Data Quality Guide for Oracle Data Quality, Siebel CRM 25.7 Update***

| Topic | Description |
|---|---|
| *External Key Generation Function* | Modified topic. Describes how Siebel CRM and Oracle Enterprise Data Quality (OEDQ) are integrated using OEDQ connector. |

## What's New in Siebel Data Quality Guide for Oracle Customer Hub, Siebel CRM 21.4 Update

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

***What's New in Siebel Data Quality Guide for Oracle Data Quality, Siebel CRM 21.4 Update***

| Topic | Description |
|---|---|
| *Siebel Data Quality Objects* | Modified topic. Removed detailed list of objects required by Siebel Data Quality. All objects in the DeDuplication project are required. |

## What's New in Siebel Data Quality Guide for Oracle Customer Hub, Siebel CRM 20.1 Update

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

# What's New in Siebel Data Quality Guide for Oracle Customer Hub, Siebel CRM 19.1 Update

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

# 2 Overview of Data Quality

## Overview of Data Quality

This chapter provides an overview of data quality functionality and products for Siebel CRM and Oracle Customer Hub. It includes the following topics:

- *Data Profiling*
- *Data Parsing and Standardization*
- *Data Matching and Data Cleansing*
- *Data Quality Products for Data Matching and Data Cleansing*
- *How Data Quality Relates to Other Entities in Siebel CRM*

**Note:** The procedures in this guide assume that you do not use the Tree navigation control option to access screens and views. However, you can choose to use the Tree navigation control if required. For more information about setting navigation options, see *Siebel Fundamentals Guide* .

## Data Profiling

Data profiling typically provides profiling capabilities that are set in an application specifically designed to give control of data quality processes to business information owners, such as data analysts and data stewards. Data profiling also provides data analysis, reporting, and monitoring capabilities.

When data quality is measured, it can be effectively managed. Data profiling provides the metrics and reports that business information owners need to continuously measure, monitor, track, and improve data quality at multiple points across the organization. Data profiling also enables business information owners and IT (information technology) to work together to deploy lasting data quality programs. Business information owners use data profiling to build data quality rules and define data quality targets together with the IT team, which then manages deployment enterprise-wide.

You can use data profiling to:

- Analyze and rank data according to completeness, conformity, consistency, duplication, integrity, and accuracy (you must use rules and reference data to analyze and rank data).
- Identify, categorize, and quantify low-quality data

For more information about data profiling and Oracle data profiling offerings, see *Oracle Fusion Middleware Upgrade Guide for Oracle Data Integrator 11g Release 1* on Oracle Technology Network (*http://www.oracle.com/technetwork/indexes/documentation/index.html*).

# Data Parsing and Standardization

Data parsing and standardization typically provides data standardization capabilities, enabling data analysts and data stewards to standardize and validate their customer data. An interface is usually included which can be used to design, build, and manage data quality efforts.

The solution offers data parsing and standardization capabilities that can be used to:

- Standardize, validate, enhance, and enrich your customer data
- Standardize and validate mailing addresses for a wide range of countries
- Parse and standardize freeform text data elements (you must use rules and reference data dictionaries to parse and standardize freeform text data elements.)

For more information about data parsing and standardization and Oracle offerings within the data parsing and standardization arena, see *Oracle Fusion Middleware Upgrade Guide for Oracle Data Integrator 11g Release 1* on Oracle Technology Network (*http://www.oracle.com/technetwork/indexes/documentation/index.html*).

# Data Matching and Data Cleansing

The data stored in account, contact, and prospect records in Oracle's Siebel CRM represents your existing and potential customers. Because of the importance of this data, maintaining its quality is essential. To ensure data quality, functionality is provided to clean this data and to remove duplicated data.

## Data Cleansing

*Data cleansing* is used to correct data and make data consistent in new or modified customer records and typically consists of the following functions:

- **Automatic population of fields in addresses.** If a user enters valid values for Zip Code, City, and Country, data quality automatically supplies a State field value. Likewise, if a user enters valid values for City, State, and Country, data quality automatically supplies a Zip Code value.
- **Address correction.** Data quality stores street address, city, state, and postal code information in a uniform and consistent format, as mandated by U.S. postal requirements. For recognized U.S. addresses, address correction provides ZIP+4 data correction and stores the data in certified U.S. Postal Service format. For example, 100 South Main Street, San Mateo, CA 94401 becomes 100 S. Main St., San Mateo, CA 94401-3256.
- **Capitalization.** Based on configuration, data quality converts fields for account, contact, prospect, and address to mixed case, all lowercase, or all uppercase.
- **Standardization.** Data quality ensures account, contact, and prospect information is stored in a uniform and consistent format. For example, IBM Corporation becomes IBM Corp.

Data cleansing is supported for the Account, Business Address, Contact, and List Mgmt Prospective Contact business components. For each business component, particular fields are used in data cleansing and this set of fields is configurable.

## Data Matching

*Data matching* is the identification of potential duplicates for account, contact, and prospect records. Potential duplicate records are displayed in the Siebel application allowing you to manually merge duplicate records into a single record.

Data matching is supported for the Account, Contact, and List Mgmt Prospective Contact business components. For each business component, a set of fields is used for comparisons in the data matching process. The set of fields is configurable, and you can also specify other matching preferences such as the degree of matching required for records to be identified as potential duplicates.

> **Tip:** The term *deduplication* is often used as a synonym for data matching particularly in names of system parameters.

In data quality you can enable and use both data cleansing and data matching at the same time, or you can use data cleansing and data matching on their own.

# Data Quality Products for Data Matching and Data Cleansing

The data quality products available for performing data quality functions within Siebel CRM enterprise and Oracle Customer Hub are divided into two categories:

- Data quality products that are embedded into Siebel CRM enterprise and Oracle Customer Hub
- Data quality products that use an open connector to connect to third-party data quality vendors

## Embedded Data Quality products

The data quality products that are embedded into Siebel CRM and Oracle Customer Hub for data matching and cleansing are:

- Oracle Data Quality Matching Server. Provides real-time and batch data matching functionality using licensed third-party Informatica Identity Resolution software with functionality from Informatica Identity Resolution. For more information, see *Oracle Enterprise Data Quality Matching Server*.
- Oracle Data Quality Address Validation Server. Provides address validation and standardization functionality using licensed third-party Informatica Identity Resolution software with functionality from Informatica Identity Resolution. For more information, see *Universal Connector*.
- Oracle Enterprise Data Quality Matching Server. Provides real-time and batch data matching functionality using licensed Oracle Enterprise Data Quality Matching Server. For more information, see *Oracle Enterprise Data Quality Matching Server*.
- Oracle Enterprise Data Quality Address Validation Server. Provides address validation and standardization functionality using licensed Oracle Enterprise Data Quality Address Validation Server. For more information, see *Oracle Enterprise Data Quality Address Validation Server*.

## Open Connector to Third-Party Data Quality Vendors

The Universal Connector provides real-time and batch data matching functionality and data cleansing functionality, as long as the associated third-party software also supports data cleansing.

**Note:** In previous releases, Universal Connector was known as SDQ Universal Connector.

If using a third-party data quality vendor for data matching, then Siebel Data Quality is mandatory (since Siebel Data Quality has the underlying infrastructure for enabling data quality). Integration between the Siebel application and the third-party data quality vendor is not possible without Siebel Data Quality.

Siebel Data Quality is a user based license, containing the underlying infrastructure and business services for enabling data quality. All Siebel CRM data quality users must license data quality at the user level using Siebel Data Quality.

### Related Topic

*Related Data Quality Objects in Siebel Tools*

# Oracle Data Quality Matching Server

The Oracle Data Quality Matching Server provides real-time and batch data matching functionality using licensed third-party IIR software.

The Oracle Data Quality Matching Server is an identity search application that searches your identity data, finds duplicates in it, and matches any duplicates found to other identity data. Running as an application server or suite of servers, Oracle Data Quality Matching Server does the following:

- Reads identity data from your databases, using specified instructions and permissions.
- Does not change your data but instead keeps a copy of it, thereby ensuring data consistency.
- Builds the SSA_NAME3 fuzzy indexes, thereby enabling the correct identity data to be found.
- Provides several simple search client procedures including, single search, batch search, and duplicate finder.

## About Using the Oracle Data Quality Matching Server

You can use theOracle Data Quality Matching Server to do the following:

- Perform real-time search for people, companies, contacts, addresses, and households.
- Discover duplicates and establish relationships in real time.
- Build relationship link tables.
- Match external files and databases.

The Oracle Data Quality Matching Server connector uses the Universal Connector in a mode where match candidate acquisition takes place within the Oracle Data Quality Matching Server, not within Siebel CRM. Since the match keys are generated and stored within the Oracle Data Quality Matching Server, key generation and key refresh operations are eliminated within Siebel CRM. This integration, whereby match candidate acquisition takes place within the Oracle Data Quality Matching Server cannot be used by other third-party data quality matching engines.

For more information about Oracle Data Quality Matching Server installation and configuration, see *Process of Installing the Oracle Data Quality Matching Server* and *Configuring Oracle Data Quality Matching Server*.

# Oracle Data Quality Address Validation Server

The Oracle Data Quality Address Validation Server is an address standardization application that provides capabilities to parse, standardize, transliterate, duplicate, and validate address data, resulting in improved address data quality. The validation capability requires the licensing of appropriate postal directories for the countries where address validation is required.

The Oracle Data Quality Address Validation Server uses a licensed version of the third-party software, IIR, for data cleansing.

Features of Oracle Data Quality Address Validation Server are:

- Integrated single API supporting all countries:

  Oracle Data Quality Address Validation Server lets you use a single API for all countries, so that you can start working immediately and add countries without the need for additional programming. The API is compatible with all major programming languages.
- Advanced validation, and correction of worldwide postal addresses, including address coverage for more than 240 countries:

  Oracle Data Quality Address Validation Server matches and corrects all address data, filters out superfluous information, assesses deliverability, and generates a detailed report with suggestions for possible sources of address problems.
- Parsing and standardization:

  Oracle Data Quality Address Validation Server parses both structured and unstructured data, identifies residues, and formats and standardizes the data (without the need for payment of special data license fees).
- Convenient updating:

  Postal reference tables in many countries change frequently. Oracle Data Quality Address Validation Server has arrangements with many local postal organizations (including Informatica Address Doctor) that allows you to receive monthly, quarterly, or biannual updates. Reference tables for each country are provided in a separate, operating system-independent database that is easy to update from a CD, DVD, or by downloading over the Internet.

The Universal Connector is integrated with the Oracle Data Quality Address Validation Server for data cleansing.

## About Using the Oracle Data Quality Address Validation Server

You can use the Oracle Data Quality Address Validation Server to cleanse data on account, contact, and prospect data from the UI in your Siebel application, or by running a batch job in Siebel CRM. You can also cleanse the data in EAI mode by sending in the address data in Simple Object Access Protocol (SOAP) format.

When you enter a new address using the contact or account screen in your Siebel application, all address data is validated, cleansed, and standardized before being committed to the Siebel database. If the address cannot be validated, then the address is standardized by using the Upper, Lower, or Camel case (depending on Oracle Data Quality Address Validation Server configuration). In addition, the account name, contact name, and other attributes are standardized.

When new contacts, accounts, or addresses are entered into Siebel CRM through a batch job, address standardization is applied before committing any records to the Siebel database.

In all cases:

- The Oracle Data Quality Address Validation Server evaluates and modifies the record according to configuration.

  Oracle Data Quality Address Validation Server returns an address validation flag and the validation status.
- The Siebel database is then updated with the cleansed data, which has been formatted and standardized with address validation.
- In the Siebel application, the updated cleansed record is displayed on the UI.

For more information about Oracle Data Quality Address Validation Server installation and configuration, see *Process of Installing the Oracle Data Quality Address Validation Server* and *Configuring Siebel CRM for the Oracle Data Quality Address Validation Server*.

# Oracle Enterprise Data Quality Matching Server

Oracle Enterprise Data Quality Match and Merge provides matching capabilities that allow you to identify matching records and optionally link or merge matched records based on deduplication rules for determining which records should remain and which should be deleted. Flexible rule configuration enables you to tune the rules to suit the task and support an iterative approach. A separate capability for simple reviews allows you to expose the match results for review without accessing the underlying rules configuration.

Used in conjunction with other Oracle Data Quality products Oracle Enterprise Data Quality Match and Merge is a flexible solution. Oracle Enterprise Data Quality Match and Merge also includes a connector that enables you to access data in Siebel CRM. Audit capabilities allow you to run data quality rules and flow-control within your data quality processes. Dashboard functionality presents results of the audit processes in graphical format, while real-time Web service functionality enables the assembled data quality process to be called as a real-time service.

# Oracle Enterprise Data Quality Address Validation Server

The Oracle Enterprise Data Quality Address Validation Server is an address standardization application that provides capabilities to parse, standardize, transliterate, duplicate, and validate address data, resulting in improved address data quality. The validation capability requires the licensing of appropriate postal directories for the countries where address validation is required. The Oracle Enterprise Data Quality Address Validation Server uses a licensed version of the third-party software, Loqate, for address cleansing.

# Universal Connector

**Note:** In previous releases, Universal Connector was known as SDQ Universal Connector.

The Universal Connector is a connector to third-party software that allows Siebel CRM to use the capabilities of a third-party application for data matching, data cleansing, or both data matching and data cleansing on account, contact, and prospect data within the Siebel application.

The Universal Connector supports data cleansing on account, contact, and prospect data in real-time and batch processing modes. The Universal Connector works across various languages and operating systems, though the

support offered by particular third-party software for data matching or data cleansing might not cover all of the languages supported by Siebel CRM. For more information about platforms supported, see the Certifications tab on My Oracle Support.

To use the Universal Connector, you must obtain, license, and install third-party software in addition to obtaining Siebel Data Quality product licensing. The data matching and data cleansing capabilities of the Universal Connector are driven by the capabilities and configuration options of the third-party software.

> **Note:** Certain third-party software from data quality vendors are certified by Oracle. For information about third-party solutions and about products that are certified for the Universal Connector, visit the Alliances section and the Partners section on the Oracle and Siebel Web site: *http://www.oracle.com/siebel/index.html*

The Universal Connector can be used in two different modes:

- The Oracle Data Quality Matching Server connector uses the Universal Connector in a mode where match candidate acquisition takes place within the Oracle Data Quality Matching Server. This mode applies only to the Oracle Data Quality Matching Server.
- Third-party data quality vendors use the Universal Connector in a mode where match candidate acquisition takes place within Siebel CRM.

You can configure the Universal Connector to specify which fields are used for data cleansing and data matching and their mapping to external application field names.

> **Note:** The Oracle Data Quality License is valid only for use with Oracle Master Data Management and Oracle CRM deployments.

# How Data Quality Relates to Other Entities in Siebel CRM

The data quality products integrate into the overall Siebel CRM environment from Oracle, as shown in the following image:

- In real-time mode, the Universal Connector is called by interactive object managers such as the Call Center object manager.
- In batch mode, the Universal Connector is called by the preconfigured server component, Data Quality Manager (DQMgr), either from the Siebel application user interface, or by starting tasks with the Siebel Server Manager command-line interface, the srvrmgr program. For more information, see *Siebel System Administration Guide* on Siebel Bookshelf.

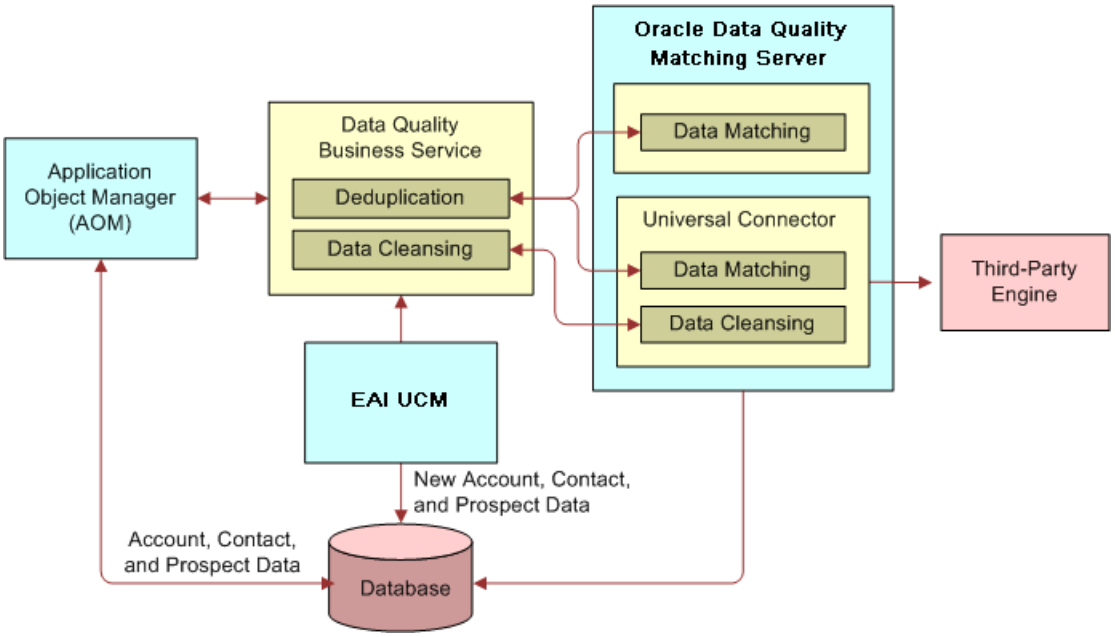
  > **Note:** The *Siebel Bookshelf* is available on Oracle Technology Network (*http://www.oracle.com/technetwork/indexes/documentation/index.html*) and Oracle Software Delivery Cloud. It might also be installed locally on your intranet or on a network location.

- The Universal Connector obtains account, contact, and prospect field data from the Siebel database using the Deduplication business service for data matching, and the Data Cleansing business service for data cleansing. Like other business services, these are reusable modules containing a set of methods. Using data quality functionality, business services simplify the task of moving data and converting data formats between the

Siebel application and external applications. The business services can also be accessed by Siebel VB or Siebel eScript code or directly from a workflow process.

- The fields used in data cleansing and data matching are sent to the appropriate cleansing or matching engine.
- Data matching and data cleansing can also be enabled for the Enterprise Application Integration (EAI) adapter and Oracle's Siebel Universal Customer Master (UCM) products.

For more information about business services and enabling data quality when using EAI, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .



# 3 Data Quality Concepts

## Data Quality Concepts

This chapter provides the conceptual information that you must use to configure data quality for Siebel CRM and Oracle Customer Hub. It includes the following topics:

- *Data Cleansing*
- *Data Matching*
- *Match Key Generation*
- *Identification of Candidate Records*
- *Calculation of Match Scores*
- *Displaying Duplicates*
- *Fuzzy Query*

## Data Cleansing

The Universal Connector supports data cleansing on the Account, Business Address, Contact, and List Mgmt Prospective Contact business components. For Siebel Industry Applications, the CUT Address business component is used instead of the Business Address business component.

> **Note:** Functionality for the CUT Address business component and Personal address business component varies. For example, only unique addresses can be associated with Contacts or Accounts when using the Personal Address. In contrast, the CUT Address does not populate the S_ADDR_PER.PER_ID table column, thereby allowing non-unique records to be created according to the S_ADDR_PER_U1 unique index and associated user key.

For each type of record, data cleansing is performed for the fields that are specified in the Third Party Administration view. The mapping between the Siebel application field names and the vendor field names is defined for each business component.

In real-time mode, data cleansing begins when a user saves a newly created or modified record. When the record is committed to the Siebel database:

1. A request for cleansing is automatically submitted to the Data Cleansing business service.
2. The Data Cleansing business service sends the request to the third-party data cleansing software, along with the applicable data.
3. The third-party software evaluates the data and modifies it in accordance with the vendor's internal instructions.
4. The third-party software sends the modified data to the Siebel application, which updates the database with the cleansed information and displays the cleansed information to the user.

In batch mode you use batch jobs to perform data cleansing on all the records in a business component or on a specified subset of those records. For data cleansing batch jobs, the process is similar to that for real-time mode, but the batch job corrects the records without immediately displaying the changes to users. The process starts when an administrator runs the server task, and the process continues until all the specified records are cleansed.

If both data cleansing and data matching are enabled, data cleansing is done first. For information about running data cleansing batch jobs, see *Cleansing Data Using Batch Jobs*.

# Data Matching

The Universal Connector and Matching Server supports data matching on the Account, Contact, and List Mgmt Prospective Contact business components. For each type of record, data matching is performed for the current record against all other records of the same type, and with the same match keys, in the application using the fields specified in the Third Party Administration view. The mapping between the Siebel application field names and the vendor field names is defined for each business component.

Data quality performs matching using fields, for example, addresses, that can have multi-value group (MVG) values associated with the type of record being matched. However, data quality is not currently able to match using MVGs. Therefore, when performing matching for a contact, data quality checks only the primary address for each contact record and does not consider other addresses.

In real-time data matching, whenever an account, contact, or prospect record is committed to the database, a request is automatically submitted to the Deduplication business service. The business service communicates with third-party data quality software, which checks for possible matches to the newly committed record and reports the results to the Siebel application.

In batch mode data matching, you first start a server task to generate or refresh the keys, and then start another server task to perform data matching. For information about performing batch mode data matching, see *Matching Data Using Batch Jobs*.

In both real-time and batch mode, whenever a primary address is updated for an account or contact record, match keys are regenerated and data matching is performed for that account or contact.

The following is the overall sequence of events in data matching:

1. Match keys are generated for database records for which data matching is enabled.
2. When a user enters or modifies a record in real-time mode, or the administrator submits a batch data matching job:
    a. A request is automatically submitted to the Deduplication business service.
    b. Using match keys, candidate matches are identified for each record. This is a means of filtering the potential matching records.
    c. The Deduplication business service sends the candidate records to the third-party software.
    d. The third-party software evaluates the candidate records and calculates a match score for each candidate record to identify the duplicate records.
    e. The third-party software returns the duplicate records to the Siebel application.
3. The duplicate records are displayed either in a window for real-time mode, or in the Administration - Data Quality views, from which you can manually merge records into a single record.

**Note:** If using theOracle Data Quality Matching Server for data matching, then you carry out deduplication against either the primary address or all address entities depending on configuration. For more information about deduplication against multiple addresses, see *Configuring Deduplication Against Multiple Addresses*.

# Match Key Generation

When data matching is performed in real time or in batch mode, data quality searches in the database for records that potentially match the current record (the record entered by a real-time user or the active record in the batch job). These records are called *candidate records*. When comparing the current records with existing records in the database, data quality does not use raw data, but instead uses *match key* values.

Match keys are calculated by applying an algorithm to specified fields in customer records. Typically keys are generated from a combination of name, address, and other identifier fields. For example, a person's name (first name, middle name, last name) for prospects and contacts, or the account name for accounts.

You generate match keys for records in the database by using batch jobs, as described in *Generating or Refreshing Keys Using Batch Jobs*.

Typically, an administrator generates and refreshes keys on a periodic basis by running batch jobs. In such batch jobs, keys can be generated for all account keys, all contact keys, all prospect keys, or subsets as defined by search specifications that include a WHERE clause.

Because key data can become out of sync with the base tables, you must refresh the key data periodically. Key generation re-generates the keys for all the records covered by the search specification. Key refresh however, only re-generates the keys for records that are new or have been modified since your last key generation, and which are covered by the search specification. Key refresh is therefore much faster than key generation.

For example, if there are records as follows:

- **Record 1.** The record has a key and has not been updated.
- **Record 2.** The record has been updated therefore the key is out of sync with the record.
- **Record 3.** The record is a new record and no key is generated for it yet.

If you generate match keys with a search specification that covers record 1, 2, and 3, new keys are generated for record 1, 2, and 3. However, if you refresh match keys with a search specification to cover record 1, 2, and 3, new keys are generated for record 2 and 3 only.

The batch capability is useful in the following circumstances:

- If you deploy data quality in a Siebel application implementation that already contains data
- If you receive new data using an input method that does not involve object manager, such as EIM or batch methods such as the List Import Service Manager
- To periodically review data to ensure the correctness of previous matching efforts.

For instructions about using batch jobs to generate or refresh keys, see *Generating or Refreshing Keys Using Batch Jobs*.

Additionally, if real-time data matching is enabled for users, keys are automatically generated (or refreshed) for a record whenever the user saves a new Account, Contact, or List Mgmt Prospective Contact record or modifies and commits an existing record to the database.

If no keys are generated for a certain record, that record is ignored as a potential candidate record when matching takes place.

# Match Key Generation with the Oracle Data Quality Matching Server

When the Universal Connector is integrated with the Oracle Data Quality Matching Server for data matching, it supports data matching on account, contact, and prospect data in real-time and batch processing modes. Whenever a record is created or updated in real-time or batch mode, match keys are generated by and stored within the Oracle Data Quality Matching Server. As a result, the information in *Match Key Generation* does not apply.

# Match Key Generation Using the Universal Connector with Third-Party Data Quality Vendors

When the Universal Connector is integrated with any other third-party data quality vendor software for data matching, match key generation is as described in this topic. That is, match keys are generated by and stored within Siebel CRM whenever a record is created or updated in real-time or in batch mode.

The Universal Connector uses one or multiple keys for each account, contact, or prospect record. The keys are calculated by reading data from specific fields in the record. The fields used depend on the business component configuration, but they can include account name, postal code, street address, or last name fields.

The value of the match keys depend on a business component-specific Dedup Token Expression parameter, as shown in the following table.

You can customize the Dedup Token Expression but it must be consistent with the internal matching logic of the vendor, which is different for each vendor. For optimal results therefore, change the values only after consulting the relevant vendor.

The generation of multiple match keys enhances the span of search for potential duplicate records, and improves match results. However, you must remember that there is a performance impact from using multiple keys.

Keys are stored in the DEDUP_TOKEN fields of the following tables:

- S_DQ_ORG_KEY (for Accounts)
- S_DQ_CON_KEY (for Contacts)
- S_DQ_PRSP_KEY (for Prospects)

You must activate the Dedup Token field in each business component in order to generate the correct match keys. If the Dedup Token field is not defined, match key generation methods will not be called. You must add the user property for the Token Expression along with the Query Expression so that the correct match keys can be generated and stored in the DEDUP_TOKEN field.

> **Note:** In Siebel CRM 7.8.x, the column DEDUP_TOKEN is available in the following tables: S_CONTACT, S_ORG_EXT, S_PRSP_CONTACT.

# Identification of Candidate Records

The way in which candidate records are identified differs for the Oracle Data Quality Matching Server and the Universal Connector as described in the following topics.

## Identification of Candidate Records with the Oracle Data Quality Matching Server

When using the Oracle Data Quality Matching Server for data matching, identification of candidate records is irrelevant as match candidate acquisition takes place within the Oracle Data Quality Matching Server.

## Identification of Candidate Records with the Universal Connector

Data quality queries the database for candidate records by using a Dedup Query Expression parameter specific to the current Business Component. A Dedup Query Expression is used rather than the related Dedup Token Expression, for the following reason: If a user does not specify a value for any of the fields that compose the Dedup Token Expression, then the token is constructed with an underscore (_) instead of a value in the part of the expression that corresponds to that field. If the token were to be used in a query, the effect would be for the query to seek records that had NULL values in corresponding fields. In contrast, the Dedup Query Expression replaces each underscore in the Dedup Token Expression with a '?' wildcard character that matches any single character, leading to the desired query results.

You can customize both the Dedup Token Expression and the Dedup Query Expression parameters through the Third Party Administration view. The configuration of these expressions must be consistent with the internal matching logic of the vendor, which is different for each vendor. For optimal results therefore, change these values only after consulting the relevant vendor. If you change the expressions, you must regenerate match keys.

See the following table for examples about how the default expressions can differ for different business components.

| Business Component | Dedup Token Expression Parameter (Key) | Dedup Query Expression Parameter (for Queries) |
|---|---|---|
| Account | "IfNull (Left ([Primary Account Postal Code], 5), '____') + IfNull (Left ([Name], 1), '_') + IfNull (Mid ([Street Address], FindNoneOf ([Street Address], '1234567890 '), 1), '_')" | "IfNull (Left ([Primary Account Postal Code], 5), '?????') + IfNull (Left ([Name], 1), '?') + IfNull (Mid ([Street Address], FindNoneOf ([Street Address], '1234567890 '), 1), '?')" |
| Contact | "IfNull (Left ([Postal Code], 5), '_____') + IfNull (Left ([Account], 1), '_') + IfNull (Left ([Last Name], 1), '_')" | "IfNull (Left ([Postal Code], 5), '?????') + IfNull (Left ([Account], 1), '?') + IfNull (Left ([Last Name], 1), '?')" |
| List Mgmt Prospective Contact | "IfNull (Left ([Postal Code], 5), '_____') + IfNull (Left ([Account], 1), '_') + IfNull (Left ([Last Name], 1), '_')" | "IfNull (Left ([Postal Code], 5), '?????') + IfNull (Left ([Account], 1), '?') + IfNull (Left ([Last Name], 1), '?')" |

The maximum number of candidate records that are sent to the third-party software at one time is determined by the value of the following vendor parameters in the Third Party Administration view:

- **Realtime Max Num of Records.** Used in real time, the default value is 200, which is the highest value that you can set. Usually there will not be more than 200 records to send, but if there are more than 200 records, the first 200 records are sent.
- **Batch Max Num of Records.** Used in batch mode, the default is 200, which is the highest value that you can set. If there are more than 200 records to send, the first 200 records are sent, then up to 200 records in the next iteration, and so on.

**Note:** Information in this topic does not apply if using the Oracle Data Quality Matching Server for data matching, because match candidate acquisition takes place within the Oracle Data Quality Matching Server.

# Calculation of Match Scores

After data quality identifies candidate records, they are sent to the third-party software. The software calculates a match score from 0 to 100 to indicate the degree of similarity between the candidate records and the current record.

The match score is calculated using a large number of rules that compensate for how frequently a given name or word appears in a language. The rules then weigh the similarity of each field on the record according to the real-world frequency of the name or word. For example, Smith is a common last name, so a match on a last name of Smith would carry less weight than a match on a last name that is rare.

The algorithms used to calculate match scores are complex. These algorithms are the intellectual property of third-party software vendors, Oracle cannot provide details about how these algorithms work.

## Calculation of Match Scores Using the Universal Connector with Third-Party Data Quality Vendors

The third-party software examines the candidate records, computes a match score for each record that is identified as a duplicate, and returns the duplicate records to data quality. The match score is a number that represents the similarity of a record to the current active record. It is calculated taking into account a large number of rules along with a number of other factors and weightings.

# Displaying Duplicates

**Note:** This applies to all data quality products.

After calculating match scores, the third-party software returns duplicate records to the Siebel application.

In real-time mode, the Siebel application displays the duplicate records in a window. These windows are:

- DeDuplication Results (Account) List Applet
- DeDuplication Results (Contact) List Applet

- DeDuplication Results (Prospect) List Applet

You can however, configure the names of these windows as described in *Configuring the Windows Displayed in Real-Time Data Matching*.

The user can either choose a record for the current record to be merged with, or click Ignore to leave the possible duplicates unchanged. For more information, see *Real-Time Data Cleansing and Data Matching*.

In batch mode, duplicate records are displayed in the Duplicate Account Resolution, Duplicate Contact Resolution, and Duplicate Prospect Resolution views in the Administration - Data Quality screen and also in the following views:

- Account Duplicates Detail View
- Contact Duplicates Detail View
- List Mgmt Prospective Contact Duplicates Detail View.

The user can then decide about which records to retain or merge with the retained records. For information about merging records, see *Merging Duplicate Records* .

If data cleansing is enabled for Siebel Universal Customer Master, you can use the following views of the Administration - Universal Customer Master screen to display duplicates:

- UCM Account Duplicates Detail View
- UCM Contact Duplicates Detail View

The default data quality views for accounts and contacts must be disabled. There is no separate UCM view for prospects.

# Fuzzy Query

Fuzzy query is an advanced query feature that makes searching more intuitive and effective. It uses fuzzy logic to enhance your ability to locate information in the database.

Fuzzy query is useful in customer interaction situations for locating the correct customer information with imperfect information. For example, fuzzy query makes it possible to find matches even if the query entries are misspelled. As an example, in a query for a customer record for Stephen Night, you can enter Steven Knight and records for Stephen Night as well as similar entries like Steve Nite are returned.

Standard query methods can rule out rows due to lack of exact matches, whereas fuzzy query does not rule out rows that contain only some of the query specifications. The fuzzy query feature is most useful for queries on account, contact, and prospect names, street names, and so on.

Fuzzy query operates as follows:

1. A user enters a query from the Siebel application GUI.
2. Data quality inspects the query for wildcard characters, such as the * (asterisk). If any wildcards are present, data quality uses standard query functionality for that query, not fuzzy query functionality.
3. Data quality generates a Dedup Token from certain specified fields in the current query input, and uses the token to query the database for possible data matches. Data quality preserves query text in fields that the DeDuplication service does not evaluate for potential data matches. For more information about Dedup Tokens, see *Identification of Candidate Records*.
4. The remainder of the process depends on the number of records that are returned in the previous step:
    - If the preliminary query results contain more records than the value of the Fuzzy Query Max Results setting, then data quality calls the DeDuplication business service, which works with the third-party data

matching engine to evaluate the possible matches. The query result returns the best available matches, up to the number of records specified by Fuzzy Query Max Results.

- If the preliminary query results contain fewer records than the value of the Fuzzy Query Max Results setting, then data quality returns all of those records as the query result, sorted according to the default sort specification for the business component.

Fuzzy query is not enabled by default; to use fuzzy query you must enable it and ensure that other conditions are met as described in *Enabling and Disabling Fuzzy Query*.

## Related Topic

*Using Fuzzy Query*

# 4 Enabling and Disabling Data Matching and Data Cleansing

## Enabling and Disabling Data Matching and Data Cleansing

This chapter describes how to enable data matching and data cleansing, and describes the data quality settings that you can apply for Siebel CRM and Oracle Customer Hub. Data cleansing and data matching must be enabled before you perform data quality tasks. This chapter includes the following topics:


- *Levels of Enabling and Disabling Data Cleansing and Data Matching*
- *Enabling Data Quality at the Enterprise Level*
- *Specifying Data Quality Settings*
- *Enabling Data Quality at the Object Manager Level*
- *Enabling Data Quality at the User Level*
- *Disabling Data Cleansing for Specific Records*
- *Enabling and Disabling Fuzzy Query*
- *Identifying Mandatory Fields for Fuzzy Query*


## Levels of Enabling and Disabling Data Cleansing and Data Matching

In Siebel CRM, there are various levels at which you can enable or disable data cleansing and data matching as summarized in the following table. In some of these views you can also specify the vendor used for data cleansing or data matching (the type). The table also shows who is most likely to set the parameters in each view.

| Screen and View | Setting or Parameter | Value | Parameters Set By |
|---|---|---|---|
| **Values set at the enterprise level** | | | |
| Administration - Server Configuration, Enterprises, Data Quality | Enable button<br><br>Disable button | Not applicable | Application administrator |
| Administration - Server Configuration, Enterprises, Parameters | DeDuplication Data Type<br><br>Data Cleansing Type | *Data Matching Vendor Name*<br><br>*Data Cleansing Vendor Name* | Application administrator |

| Screen and View | Setting or Parameter | Value | Parameters Set By |
|---|---|---|---|
| **Values set for Data Quality Settings**<br><br>**Note:** These settings affect all the servers. | | | |
| Administration - Data Quality, Data Quality Settings | Enable DataCleansing<br><br>Enable DeDuplication | Yes or No | Data administrator |
| **Values set at the object manager level** | | | |
| Administration - Server Configuration, Servers, select component Data Quality Manager, then click the Parameters tab | Data Cleansing Enable Flag<br><br>Data Cleansing Type<br><br>DedDuplication Enable Flag<br><br>DeDuplication Data Type | True or False<br><br>*Data Cleansing Vendor Name*<br><br>True or False<br><br>*Data Matching Vendor Name* | Data administrator |
| Administration - Server Configuration, Servers, select Object manager of application (for example, Sales Object Manager (ENU)), then click the Parameters tab | Data Cleansing Enable Flag<br><br>Data Cleansing Type<br><br>DedDuplication Enable Flag<br><br>DeDuplication Data Type | True or False<br><br>*Data Cleansing Vendor Name*<br><br>True or False<br><br>*Data Matching Vendor Name* | Data administrator |
| **Values set at the user level** | | | |
| Tools, User Preferences, Data Quality | Enable DataCleansing<br><br>Enable DeDuplication | Yes or No | Data steward and end users<br><br>**Note:** A data steward monitors the quality of incoming and outgoing data for an organization. |

The values of parameters at the user level override the values at the object manager level. In turn, the values at the in the object manager level override the settings specified at the enterprise level. This allows administrators to enable data matching or cleansing for one application but not another and allows users to disable data matching or cleansing for their own login even if data matching or cleansing is enabled for their application.

However, data matching or data cleansing cannot be enabled for a user login if data matching or data cleansing are not enabled at the object manager level.

Even if data cleansing and data matching are enabled, cleansing and matching are only triggered for business components as defined in Siebel Tools and in the Data Quality - Administration views.

For more information, see the following topics:

- *Enabling Data Quality at the Enterprise Level*
- *Specifying Data Quality Settings*
- *Enabling Data Quality at the Object Manager Level*
- *Enabling Data Quality at the User Level*

# Enabling Data Quality at the Enterprise Level

Before performing any batch data matching or date cleansing tasks, you must first enable the Data Quality Manager server component for the enterprise. Data Quality Manager is the preconfigured component in the Data Quality component group that you use to run your data quality tasks.

There are three possible ways to enable the Data Quality component group:

- When you install a Siebel Server, you can specify the Data Quality component group in the list of component groups that you want to enable.
- If you do not choose to enable the Data Quality component group during installation, you can enable it later using the Siebel Server Manager. For more information about enabling component groups using the Siebel Server Manager, see *Siebel System Administration Guide* .
- You can enable the Data Quality component group from your Siebel application, as described in this topic.

> **Note:** If you use Siebel Server Manager (srvrmgr) to list component groups, groups that were enabled from the Siebel application are not listed.

The enterprise parameters DeDuplication Data Type and Data Cleansing Type specify respectively the type of software used for data matching and data cleansing. These parameters are automatically set according to what you choose for data matching at Siebel Server installation time. However, it is recommended that you check the values for these parameters to make sure they are appropriately set for the enterprise.

Use the following procedures to enable and disable Data Quality Manager and to configure the enterprise parameter settings for data matching and data cleansing.

## To enable data quality at the enterprise level

1. Log in to the Siebel application with administrator responsibilities.
2. Navigate to the Administration - Server Configuration screen, then Enterprises view.
3. Click the Component Groups view tab.
4. In the Component Groups list, select Data Quality, and then click the Enable button.

   Data quality is now enabled at the enterprise level for data matching and data cleansing.
5. Restart the Siebel Server.

Use the following procedure to configure data matching and data cleansing settings at the enterprise level.

# Configuring Data Matching and Data Cleansing Settings at the Enterprise Level

Use the following task to configure data matching and data cleansing settings at the enterprise level.

## To configure data matching and data cleansing settings at the enterprise level

1. Log in to the Siebel application with administrator responsibilities.
2. Navigate to the Administration - Server Configuration screen, then Enterprises view.
3. Click the Parameters view tab.
4. In the Parameter field in the Enterprise Parameters list, query and review the settings for each of the following parameters:
   - DeDuplication Data Type
   - Data Cleansing Type

   The Value field can be set as follows:
   - **CHANGE_ME.** Indicates that you chose None when you installed the Siebel Server.
   - ***name of third-party server*.** Indicates the name of the third-party server that is being used for data matching and (or) data cleansing.

     For example:
     - **ISS.** Indicates that Oracle Data Quality Matching Server is used for data matching.
     - **ASM.** Indicates that Oracle Data Quality Address Validation Server is used for data cleansing.

   If necessary, enter any corrections in the Value field.

   The value you choose for Data Cleansing Type can differ from the value you choose for DeDuplication Data Type, provided that you have the appropriate vendor software available.

   > **Note:** The values set in the Value field in the Enterprise Parameters list also appear in the Value fields for the corresponding parameters in the Component Parameters and Server Parameters views.

5. If you change an enterprise parameter in Step 4 (or if you change any value of a server component such as Data Quality Manager), restart the server component so that the new settings take effect.

   For more information about restarting server components, see *Siebel System Administration Guide* .

# Specifying Data Quality Settings

Before performing any data matching or cleansing tasks, you must make sure that the appropriate data quality setting parameters are specified. Use the following procedure to specify the data quality settings for the enterprise.

## To specify data quality settings

1. Navigate to the Administration - Data Quality screen, then the Data Quality Settings view.

2. In the Value field for each parameter, apply the appropriate settings.

   The parameters applicable to all data quality products are described in the following table.
3. Log out of the application and log back in for the changes to take effect.

> **Note:** You do not have to restart the Siebel Server.

The following table describes the parameters that apply to all data quality products.

| Parameter | Description |
|---|---|
| Enable DataCleansing | Determines whether real-time data cleansing is enabled for the Siebel Server the administrator is currently logged into. The default value is Yes. Other values you set for data quality can override this setting. For more information about this, see *Levels of Enabling and Disabling Data Cleansing and Data Matching*. |
| Enable DeDuplication | Determines whether real-time data matching is enabled for the Siebel Server the administrator is currently logged into. The default value is Yes. Other values you set for data quality can override this setting. For more information about this, see *Levels of Enabling and Disabling Data Cleansing and Data Matching*. |
| Force User Dedupe - Account | Determines whether duplicate records are displayed in a window when a user saves a new account record. The user can then merge duplicates. If set to No, duplicates are not displayed in a window, but the user can merge duplicates in the Duplicate Accounts view. The default value is Yes. For more information about window configuration, see *Configuring the Windows Displayed in Real-Time Data Matching*. |
| Force User DeDupe - Contact | Determines whether duplicate records are displayed in a window when a user saves a new contact record. The user can then merge duplicates. If set to No, duplicates are not displayed in a window, but the user can merge duplicates in the Duplicate Contacts view. The default value is Yes. For more information about window configuration, see *Configuring the Windows Displayed in Real-Time Data Matching*. |
| Force User DeDupe - List Mgmt | Determines whether duplicate records are displayed in a window when a user saves a new prospect record. The user can then merge duplicates. If set to No, duplicates are not displayed in a window, but the user can merge duplicates in the Duplicate Prospects view. The default value is Yes. For more information about window configuration, see *Configuring the Windows Displayed in Real-Time Data Matching*. |
| Fuzzy Query Enabled | Determines whether fuzzy query, an advanced search feature, is enabled. The default value is no. For more information about fuzzy query, see *Enabling and Disabling Fuzzy Query*. |
| Fuzzy Query - Max Returned | Specifies the maximum number of records returned when a fuzzy query is performed. The default value is 500. For more information about fuzzy query, see *Enabling and Disabling Fuzzy Query*. |
| Account Match Against<br>**Note:** This parameter applies to the Oracle Data Quality Matching Server only. | If set to Primary Address, then only the primary address associated with an account is considered for deduplication. If set to All Address, then all addresses associated with an account are considered for deduplication. The default value is Primary Address. |

| Parameter | Description |
|---|---|
| Contact Match Against<br><br>**Note:** This parameter applies to the Oracle Data Quality Matching Server only. | If set to Primary Address, then only the primary address associated with a contact is considered for deduplication. If set to All Address, then all addresses associated with a contact are considered for deduplication. The default value is Primary Address. |
| Match Threshold | Specifies a threshold above which any record with a match score is considered a match. Higher scores indicate closer matches. A perfect match is equal to 100. Possible values are: 50-100. |
| Enable DQ Multiple Addresses<br><br>**Note:** This parameter applies to the Oracle Data Quality Matching Server only. | Set to Yes if configuring deduplication against multiple addresses. The default value is No. For more information, see *Configuring Deduplication Against Multiple Addresses*. |
| Enable DQ Multiple Languages<br><br>**Note:** This parameter applies to the Oracle Data Quality Matching Server only. | Set to Yes if configuring multiple language support for data matching. The default value is No. For more information, see *Configuring Multiple Language Support for Data Matching*. |
| Enable DQ Sync<br><br>**Note:** This parameter applies to the Oracle Data Quality Matching Server only. | Set to Yes if configuring data synchronization between the Siebel application and the Oracle Data Quality Matching Server using the new synchronizer process. The new synchronizer process uses the DQ Sync Services business service to insert synchronized messages directly into the Oracle Data Quality Matching Server (Informatica Identity Resolution) NSA table, and is triggered by the DQ Sync* action sets in Siebel CRM. The default value is Yes.<br><br>For the new synchronizer process to work, you must also:<br><br>• Configure the EBC table. For more information, see *Process of Configuring Data Synchronization Between Siebel CRM and Oracle Data Quality Matching Server*.<br>• Activate the DQ Sync Action Sets.<br><br>**Note:** The old synchronizer uses workflows to send XML messages to the Oracle Data Quality Matching Server XS Server (XML Sync Server), and is triggered by the ISSSYNC action sets in Siebel CRM. |
| Sort Match Web Service Results | Set to Yes to enable the sort filter for the results in the Data Quality Web Services. The default is No. |

# Disabling Data Matching and Cleansing Without Restarting the Siebel Server

If you enabled data matching or cleansing from the Administration - Server Configuration screen, you can disable one or both from the Data Quality Settings view without restarting the Siebel Server.

After you disable data matching or data cleansing, log out and then log in to the application again for the new settings to take effect. The settings apply to all the object managers in your Siebel Server, whether or not they have been enabled in the Administration - Server Configuration screen.

# Enabling Data Quality at the Object Manager Level

In real-time mode, data quality is called when a new or modified record is saved. Real-time data matching and cleansing is supported only for employee-facing applications. By specifying data matching and cleansing parameters at the object manager level in the Siebel application, you can enable data matching or cleansing for one application and disable it for another application. However, you cannot enable data matching for both the Matching Server and the Universal Connector for the same application.

To enable data matching and data cleansing for real-time processing at the object manager level, you must enable certain parameters for the object manager that the application uses. You enable real-time processing for data matching and cleansing using either the graphical user interface (GUI) of the Siebel application or the command-line interface of the Siebel Server Manager.

**Note:** The command-line interface of the Siebel Server Manager is the srvrmgr program. For more information about using the command-line interface, see *Siebel System Administration Guide* .

Use the following procedures to enable data matching and cleansing for real-time processing:

- *Enabling Data Quality Using the GUI*
- *Enabling Data Quality Using the Command-Line Interface*

These procedures require that data quality is already enabled at the enterprise level. For information about enabling data quality at the enterprise level, see *Enabling Data Quality at the Enterprise Level*.

## Enabling Data Quality Using the GUI

To enable data quality at the object manager level using the GUI, complete the steps in the following procedure.

### To enable data quality at the object manager level using the GUI

1. Log in to the Siebel application with administrator responsibilities.
2. Navigate to the Administration - Server Configuration screen, then the Servers view.
3. In the Components list, select an object manager where end users enter and modify customer data.

   For example, select the Call Center Object Manager (ENU) if you want to enable or disable real-time data matching or cleansing for that object manager.
4. Click the Parameters subview tab.

5. In the Parameters field in the Component Parameters list, apply the appropriate settings to the parameters listed in the following table to enable or disable data matching or cleansing.

| Field | Description |
|---|---|
| Data Cleansing Enable Flag | Indicates whether real-time data cleansing is enabled for a specific object manager, such as Call Center Object Manager (ENU). This parameter allows you to set different data cleansing values in different object managers. By default, all values for this parameter are set to False. |
| DeDuplication Enable Flag | Indicates whether real-time data matching is enabled for a specific object manager, such as Call Center Object Manager (ENU). This parameter allows you to set different data matching values in different object managers. By default, all values for this parameter are set to False. |
| Data Cleansing Type | Indicates the third-party vendor software that is used for data cleansing. |
| DeDuplication Data Type | Indicates the third-party vendor software that is used for data matching. |

> **Note:** The settings at this object manager level override the enterprise-level settings.

6. After the component parameters are set, restart the object manager either by using srvrmgr or by completing the following sub-steps:
   a. Navigate to the Administration - Server Management screen, then the Servers view.
   b. Click the Components Groups view tab (if not already active).
   c. In the Servers list (upper applet), select the appropriate Siebel Server (if you have more than one in your enterprise).
   d. In the Components Groups list (middle applet), select the component of your object manager, and use the Startup and Shutdown buttons to restart the component.

For information about restarting server components, see *Siebel System Administration Guide* .

# Enabling Data Quality Using the Command-Line Interface

Use the following procedure to enable data quality at the object manager level using the Siebel Server Manager command-line interface.

## To enable data quality at the object manager level using the Siebel Server Manager command-line interface

1. Start the Siebel Server Manager command-line interface (srvrmgr) using the user name and password of a Siebel application administrator account such as SADMIN. For more information, see *Siebel System Administration Guide* .

> **Note:** You must have Siebel CRM administrator responsibility to start or run Siebel Server tasks using the Siebel Server Manager command-line interface.

2. Execute commands similar to the following examples to enable or disable data matching or data cleansing.
   The examples are for the Call Center English application (where SSCObjmgr_enu is the alias name of the English Call Center object manager of the Call Center application.) Use the appropriate alias_name for the application component name to which you want the change applied:
   - To enable data matching if you are using Universal Connector third-party software:

     ```
     change parameter DedDupTypeEnable=True, DeDupTypeType=ISS for component
     SCCObjMgr_enu
     ```

   - To enable data cleansing if you are using Universal Connector third-party software:

     ```
     change parameter DataCleansingEnable=True, DataCleansingType=ASM for component
     SCCObjMgr_enu
     ```

To disable data matching or data cleansing, execute commands like these examples with the DeDupTypeEnable or DataCleansingEnable parameters set to False. For more information about using the command-line interface, see *Siebel System Administration Guide* .

# Enabling Data Quality at the User Level

Users can disable data matching, data cleansing, or fuzzy query for their own logins by setting user preferences even if these features are enabled for their application. The values in the User Preferences view are applicable to real-time processing.

The User Profile screen, Data Quality view displays many of the same options that are set in the Administration - Data Quality Settings screen. However, a choice to disable a feature in the user preference settings takes priority (for the current user) over a choice to enable it in the Data Quality Settings view. The reverse is not true: if a feature is disabled in the Data Quality Settings view, you cannot override that disabling by enabling the feature in the user preferences settings.

Use the following procedure to set user preferences and enable data quality at the user level.

## To enable data quality at the user level

1. Log in to your Siebel application as the user.
2. Navigate to the User Profile screen, then the Data Quality view.
3. In the Data Quality form, set the parameters for that user.
   The following table describes the fields.

| Field | Description |
|---|---|
| Enable Data Cleansing | Select Yes to enable data cleansing for the current user. Otherwise, select No to disable data cleansing. |
| Enable DeDuplication | Select Yes to enable data matching for the current user. Otherwise select No to disable data matching. |

| Field | Description |
|---|---|
| Fuzzy Query Enabled | Select Yes to use a fuzzy query for the current user. Select No to disable fuzzy queries for the current user. Fuzzy query only works if certain conditions are met; see *Enabling and Disabling Fuzzy Query*. |
| Fuzzy Query - Max Matches Returned | Specify the maximum number of query result records you want data quality to return to you. Valid values are 10 to 500. The default value is 100. |
| Match Threshold | Applicable for Universal Connector. Select a threshold above which any record with a match score is considered a match. Higher scores indicate closer matches (a perfect match is equal to 100). Possible values are: 50-100. If no threshold value is supplied in any of the data quality settings, the default value of 50 is used by the Siebel application. |

4. Log out of the application and log back in as the user to initialize the new settings.
   Do not restart the Siebel Server.

# Disabling Data Cleansing for Specific Records

You can disable data cleansing for accounts, contacts, and prospects on a record-by-record basis for both real-time and batch mode processing.

## To disable data cleansing for a record

1. Drill down on the record for which you want to disable cleansing, and then click the More Info view tab.
2. In the More Info form, select the Disable Cleansing check box.

**Note:** The Disable Cleansing check box is cleared (that is, cleansing enabled) by default for new records.

# Enabling and Disabling Fuzzy Query

Data Quality provides an advanced query feature, known as fuzzy query, that makes searching more intuitive and effective. For general information about fuzzy query functionality, see *Fuzzy Query*.

When all of the following conditions are satisfied, the Siebel application uses fuzzy query mode automatically, regardless of which data quality product you are using. However, if any of the conditions are not satisfied, the Siebel application uses the standard query mode:

- Data matching must be enabled in the Administration - Data Quality Settings view, see *Specifying Data Quality Settings*.
- Data matching must not be disabled for the current user in the User Preferences - Data Quality view, see *Enabling Data Quality at the User Level*.

- Fuzzy query must be enabled in the Administration - Data Quality Settings view; Fuzzy Query Enabled must be set to Yes.
- Fuzzy query must be enabled for the current user in the User Preferences - Data Quality view; Fuzzy Query Enabled must be set to Yes.
- The query must not use wildcards.
- The query must specify values in fields designated as fuzzy query mandatory fields. For information about identifying the mandatory fields, see *Identifying Mandatory Fields for Fuzzy Query*.
- The query must leave optional fields blank.

The following procedures describe how to enable and disable fuzzy query in the Data Quality Settings. If wildcards (*) or quotation marks (") are used in a fuzzy query, then that fuzzy query will not be effective. Also, if mandatory fuzzy query fields are missing, then fuzzy query is disabled for that particular query.

# Enabling Fuzzy Query

Use the following procedure to enable fuzzy query.

## To enable fuzzy query

1. Navigate to the Administration - Data Quality screen, then the Data Quality Settings view.
2. Click New to create a new record:
    a. In the Name field, choose Fuzzy Query Enabled.
    b. In the Value field, choose Yes.
3. (Optional) If you want to set a maximum number of returned records, click New to create a new record:
    a. In the Name field, choose Fuzzy Query - Max Returned.
    b. In the Value field, enter a number from 10 to 500.

# Disabling Fuzzy Query

Use the following procedure to disable fuzzy query.

## To disable fuzzy query

1. Navigate to the Administration - Data Quality screen, then the Data Quality Settings view.
2. In the Data Quality Settings list, select Fuzzy Query Enabled, and in the Value field, choose No.

## Related Topics

*Using Fuzzy Query*

*Example of Enabling and Using Fuzzy Query with Accounts*

# Identifying Mandatory Fields for Fuzzy Query

You might want to provide users with information about mandatory fields (query fields that must include values for the Siebel application to use fuzzy query mode). The following table shows the preconfigured mandatory fields that Oracle provides.

If you want to identify the current mandatory fields for your own Siebel CRM implementation, use the procedure that follows.

| Business Component | Mandatory Fields for Fuzzy Query |
|---|---|
| Account | Name |
| Contact | First Name, Last Name |
| List Mgmt Prospective Contact | First Name, Last Name |

## To identify fields that are mandatory for fuzzy query

1. Login to Web Tools.
2. In the Object Explorer, expand Business Component and then select the business component of interest in the Business Components pane.
3. In the Object Explorer, select Business Component User Prop.

   > **Tip:** If the Business Component User Prop object is not visible in the Object Explorer, you can enable it in the Development Tools Options dialog box (View, Options, Object Explorer). If this is necessary, you must repeat Step 2 of this procedure.

4. In the Business Component User Properties pane, select Fuzzy Query Mandatory Fields, and inspect the field names listed in the Value column.

   Repeat Step 2 through Step 4 for other business components, as needed.

# 5 Configuring Data Quality with Universal Connector

## Configuring Data Quality with Universal Connector

This chapter describes the data quality configuration that you can perform for Siebel CRM and Oracle Customer Hub. It covers the following topics:

- *Data Quality Configuration Overview*
- *Installing the Universal Connector*
- *Process of Configuring New Data Quality Connectors*
- *Configuring Vendor Parameters*
- *Mapping of Vendor Fields to Business Component Fields*
- *Configuring the Windows Displayed in Real-Time Data Matching*
- *Configuring the Mandatory Fields for Fuzzy Query*
- *Data Quality User Properties*
- *Optimizing Data Cleansing Performance*
- *Optimizing Data Matching Performance*

**Note:** You must be familiar with Siebel Tools before performing some of the data quality configuration tasks. For more information about Siebel Tools, see *Using Siebel Tools* and *Configuring Siebel Business Applications* .

## Data Quality Configuration Overview

The following table summarizes the data quality configuration that you can perform for Siebel CRM and Oracle Customer Hub.

| Type | Configuration | See... |
|---|---|---|
| Generic data quality configuration for all data quality products | Configure new connectors for data matching and data cleansing for the Universal Connector | *Process of Configuring New Data Quality Connectors* |
| | Configure vendor parameters. You can configure the parameters for each of the software vendors. | *Configuring Vendor Parameters* |
| | Configure field mappings for business components. You can change or add field mappings. | *Mapping of Vendor Fields to Business Component Fields* |

| Type | Configuration | See... |
|---|---|---|
| | | |
| | Configure the windows displayed in real-time data matching | *Configuring the Windows Displayed in Real-Time Data Matching* |
| | Configure the mandatory fields for fuzzy search. | *Configuring the Mandatory Fields for Fuzzy Query* |
| Oracle Data Quality Matching Server Configuration | Data Matching | *Configuring Oracle Data Quality Matching Server* |
| | | *Configuring a New Field for Real-Time Data Matching* |
| | | *Loading Data Incrementally* |
| Oracle Data Quality Address Validation Server Configuration | Data Cleansing | *Configuring Siebel CRM for the Oracle Data Quality Address Validation Server* |

# Installing the Universal Connector

To use the Universal Connector, you must install the Data Quality Connector component when running the Siebel CRM installer for Siebel Server Enterprise. For installation instructions, including instructions on creating the installation image, see *Siebel Installation Guide* . For information about installing the Universal Connector on a network, see *Installing Third-Party Application Software for Use with the Universal Connector*.

**Note:** If using the Data Quality Applications product media pack on Oracle Software Delivery Cloud to install data quality products, then the Universal Connector component is installed as part of that installation process.

## Installing Third-Party Application Software for Use with the Universal Connector

Unlike most other third-party software, you must install third-party software for use with the Universal Connector after you install Siebel CRM. Install the third-party software in the SDQConnector directory where Siebel CRM is installed; that is, the *Siebel_Server_root*/SDQConnector directory. See the documentation provided by the third-party vendor for instructions.

# Installing Third-Party Data Cleansing Files for Use with the Universal Connector

To perform data cleansing, the third-party vendor software usually needs a set of files for standardization and data cleansing. For information about specifying the location of such files, see the documentation provided by the third-party vendor.

# Universal Connector Libraries

The Universal Connector uses standard Siebel CRM business services for data matching and cleansing. These business services call a generalized adapter that can communicate with an external data quality application through a set of library files.

The names of the shared libraries are vendor-specific, but must follow naming conventions as described in *About the Universal Connector API*.

The Siebel CRM installation process copies these DLL or shared library files to a location that depends on the operating system you are using, as shown in the following table.

| Does Vendor Support Multiple Languages? | DLL Storage Locations (Windows) | Shared Library Storage Locations (UNIX) |
|---|---|---|
| No | For Siebel Server:<br>`Siebel_Server_root\bin\`<br>For Developer Web Client:<br>`Client_root\bin\` | For Siebel Server:<br>`Siebel_Server_root/lib` |
| Yes | For Siebel Server:<br>`Siebel_Server_root\bin\langua ge_code`<br>For Developer Web Client:<br>`Client_root` | For Siebel Server:<br>`Siebel_Server_root/lib/language_code` |

| Does Vendor Support Multiple Languages? | DLL Storage Locations (Windows) | Shared Library Storage Locations (UNIX) |
|---|---|---|
| | `\bin\`<br>`language_code`<br><br>where language_code is the appropriate language code, such as ENU for U.S. English. | |

**Note:** The DLLs or shared libraries for each vendor can be specific to certain operating systems or external product versions, so it is important that you confirm with your vendor that you have the correct files installed on your Siebel Server.

The Universal Connector requires that you install third-party applications on each Siebel Server that has the object managers enabled for data quality functionality. If you plan to test real-time mode using a Siebel Developer Web Client, you must install the third-party Data Quality software on that computer, as well.

**Note:** When installing data quality products using the Data Quality Applications product media pack on Oracle Software Delivery Cloud, the DLL or shared library files are copied to a location that depends on the operating system you are using.

# Process of Configuring New Data Quality Connectors

You can define your own connectors for data matching and data cleansing for the Universal Connector. To configure new data quality connectors, perform the following tasks:

1. *Registering New Data Quality Connectors*
2. *Configuring Business Components and Applets for Data Matching and Data Cleansing*

**Note:** These processes do not cover vendor-specific configuration. You must work with Oracle-certified alliance partners to enhance data quality features for your applications.

## Registering New Data Quality Connectors

Data Quality connector definitions are configured in the Third Party Administration view. You can specify one external application for data matching and a different application for data cleansing for the Universal Connector. You do this by setting the correct input values for each external application.

**Note:** The vendor parameters in the Siebel application are specifically designed to support multiple vendors in the Universal Connector architecture without the need for additional code. The values of these parameters must be provided by third-party vendors. Typically, these values cannot be changed because specific values are required by each software vendor. For more information about the values to use, see the installation documentation provided by your third-party vendor.

The Deduplication and Data Cleansing business services include a generalized adapter that communicates with the external data quality application through a set of dynamic-link library (DLL) or shared library files.

The DLL Name setting in the Third Party Administration view tells the Siebel application how to load the DLL or shared library. The names of the libraries are vendor-specific, but must follow naming conventions as described in *About the Universal Connector API*. The Siebel application loads the libraries from the locations described in *Levels of Enabling and Disabling Data Cleansing and Data Matching*.

Use the following procedure to register a data quality connector. This topic is a step in *Process of Configuring New Data Quality Connectors*.

## To register a data quality connector

1. Navigate to the Administration - Data Quality screen, then the Third Party Administration view.
2. In the Vendor List, create a new record and complete the necessary fields, as shown in the following table.

| Name | Library Name |
|---|---|
| The name of the vendor.<br><br>For a data matching connector, the name must match the value specified in the server parameter DeDuplication Data Type.<br><br>For a data cleansing connector, the name must match the value specified in the server parameter Data Cleansing Type. | The name of the vendor DLL or shared library. |

# Configuring Business Components and Applets for Data Matching and Data Cleansing

This topic describes how to configure business components and applets, whether existing ones or new ones you create, for data matching and data cleansing.

You can configure existing business components or create additional business components for data matching for the Matching Server and for data matching and data cleansing for the Universal Connector.

Typically, you configure existing business components; however, you can create your own business components to associate with connector definitions. For information about how to create new business components and define user properties for those components, see *Configuring Siebel Business Applications* .

**Note:** You must base new business components you create only on the CSSBCBase class to support data cleansing and data matching, or make sure that the business component uses a class whose parent is CSSBCBase. This class includes the specific logic to call the DeDuplication and Data Cleansing business services.

To configure business components for data matching and data cleansing, complete the steps in the following procedure. This topic is a step in *Process of Configuring New Data Quality Connectors*.

## To configure business components for data matching and cleansing

1. Associate the business component with a connector.
   This includes configuring the vendor parameters shown in the following table.

| | Name | Value |
|---|---|---|
| For data matching | Business_component_name DeDup Record Type | *Business_component_name* |
| For data matching | Business_component_name Token Expression | Consult the vendor for the value of this field.<br><br>**Note:** Applies to the Universal Connector, where key generation is carried out by the Siebel application. |
| For data matching | Business_component_name Query Expression | Consult the vendor for the value of this field.<br><br>**Note:** Applies to the Universal Connector, where key generation is carried out by the Siebel application. |
| For data matching | Parameter 1 | "global", "iss-config-file", "ssadq_cfg.xml"<br><br>**Note:** Applies to the Oracle Data Quality Matching Server only, where match keys are generated by the Oracle Data Quality Matching Server. |
| For data cleansing | Business_component_name DataCleanse Record Type | *Business_component_name* |

2. Configure the field mappings for each business component and operation.
3. Create a DeDuplication Results business component and add it to the Deduplication business object.
4. Configure an applet as the DeDuplication Results List Applet.
5. Configure Duplicate views and add them to the Administration - Data Quality screen.
6. Add the business component user properties as shown in the following table.

| Property | Value |
|---|---|
| DeDuplication Results BusComp | The buscomp that you created in *Configuring Business Components and Applets for Data Matching and Data Cleansing*. |

| Property | Value |
|---|---|
| DeDuplication Results List Applet | The applet that you created in *Configuring Business Components and Applets for Data Matching and Data Cleansing*. |

7. Add a field called Merge Sequence Number to the business component and a user property called Merge Sequence Number Field.

# Configuring Vendor Parameters

For each of the third-party software vendors that data quality uses for data cleansing or data matching, you can configure the vendor parameters that the Siebel application passes to the vendor software. You configure the vendor parameters in the Administration - Data Quality screen, Third Party Administration view, which also contains the DLL or shared library name for each vendor.

There are preconfigured vendor parameters for the Universal Connector with Oracle Data Quality Matching Server and Oracle Data Quality Address Validation Server as examples.

## To configure vendor parameters

1. Navigate to the Administration - Data Quality screen, then the Third Party Administration view.
2. In the Vendor List, select the record for the required vendor.
3. Click the Vendor Parameters view tab.
4. In the Vendor Parameters List, create new records as required, or configure the values of existing vendor parameters.

# Mapping of Vendor Fields to Business Component Fields

For each vendor who supports data cleansing or data matching, there are field mappings that specify:

- The fields that are used in data cleansing and data matching
- The mapping between the Siebel application field names and the corresponding vendor field names.

There are mappings for each supported business component and data quality operation (DeDuplication and Data Cleansing). There are preconfigured field mappings for the Oracle Data Quality Matching Server Connector, see *Preconfigured Field Mappings for Oracle Data Quality Matching Server*.

You can configure the field mappings for a business component to include new fields or modify them to map to different fields. There might also be additional configuration required for particular third-party software.

**Note:** You must contact the specific vendor for the list of fields that the vendor supports for data cleansing and data matching and to understand the effect of changing field mappings.

## Related Topics

*Mapping Data Matching Vendor Fields to Siebel Business Components*

*Adding a Field Mapping for Data Matching*

*Mapping Data Cleansing Vendor Fields to Siebel Business Component Fields*

# Mapping Data Matching Vendor Fields to Siebel Business Components

This topic explains how to map data matching vendor fields to Siebel business component fields.

### To map a data matching vendor field to a Siebel business component field

1. Navigate to the Administration - Data Quality screen, then the Third Party Administration view.
2. In the Vendor List, select the record for the required vendor.
3. Click the BC Vendor Field Mapping view tab.
4. In the BC Operation list, select the record for the required business component and the DeDuplication operation. The field mappings are displayed in the Field Mapping list.
5. In the Field Mapping list enter the required values for Business Component Field and Mapped Field.

# Adding a Field Mapping for Data Matching

In addition to the preconfigured fields that are used in data matching, you can configure your data quality implementation to inspect certain additional fields during data matching, such as a date of birth field for Contacts, or a D-U-N-S number field for Accounts.

For the Universal Connector, if the key token expression changes, you must regenerate match keys. Therefore, if you are adding a new field and the new field is added to the token expression, you must generate the match keys.

The following procedure describes how to add a field mapping.

### To add a field mapping for data matching

1. Navigate to the Administration - Data Quality screen, then the Third Party Administration view.
2. In the Vendor List, select the record for the required vendor.
3. Click the BC Vendor Field Mapping view tab.
4. In the BC Operation list, select the record for the required business component and operation
   - For example, to include a date of birth as a matching criterion, select the record for Contact and DeDuplication.
   - For example, to include a D-U-N-S number as a matching criterion, select the record for Account and DeDuplication.

   The field mappings are displayed in the Field Mapping list.

5. In the Field Mapping list, create a new record and complete the necessary fields as in the example shown in the following table.

| Business Component Field | Mapped Field |
|---|---|
| Birth Date | `Contact.Birth Date` |
| DUNS Number | `Account.DUNS Number` |

6. If required, modify the corresponding real-time and batch mode data flows to incorporate the new field so that data quality considers the new field during data matching comparisons.

# Mapping Data Cleansing Vendor Fields to Siebel Business Component Fields

Data cleansing is triggered when a record is saved after a field that is defined as an active data cleansing field is updated.

Default settings are preconfigured for the Account, Contact, Prospect, and Business Address business components to support integration with Oracle Data Quality Address Validation Server, but you can configure the mappings to your requirements or to support integration to other vendors.

**Note:** For Siebel Industry Applications, the CUT Address business component is enabled for data cleansing rather than the Business Address business component.

For example the following are active data cleansing fields for the Contact business component:

- Last Name
- First Name
- Middle Name
- Job Title

**Tip:** Only fields that are preconfigured as data cleansing fields in the vendor properties trigger real-time data cleansing when they are modified.

## To map a data cleansing vendor field to a Siebel business component field

1. Navigate to the Administration - Data Quality screen, then the Third Party Administration view.
2. In the Vendor List, select the record for the required vendor.
3. Click the BC Vendor Field Mapping view tab.
4. In the BC Operation list, select the record for the required business component and Data Cleansing operation.

   The field mappings are displayed in the Field Mapping list.
5. In the Field Mapping list enter the required values for Business Component Field and Mapped Field.

# Configuring the Windows Displayed in Real-Time Data Matching

In real-time data matching when the user saves a new account, contact, or prospect record, the Siebel application displays the duplicate records in a window.

You can change the name of the windows that are displayed, and you can specify that a window is displayed for some other applets. This can be a similar applet to the Contact List, Account List, or List Mgmt Prospective Contact List applet or a customized applet. Both list and detail applets are supported, as long as they are not child applets.

For more information about configuring the windows displayed in real-time data matching, see the following procedures:

- *Changing a Window Name*
- *Adding a Deduplication Window for an Applet*
- *Configuring a Real-Time Deduplication Window for Child Applets*

## Changing a Window Name

Use the following procedure to change the name of a window displayed.

### To change the name of the window displayed

1. Login to Web Tools, open workspace.
2. In the Object Explorer, select the Applet, and then select the applet of interest, for example, Contact List Applet.
3. In the Object Explorer, select Applet User Prop.
4. Select the DeDuplication Results Applet user property and change its value as required.
5. Save the changes and submit the workspace for delivery.

## Adding a Deduplication Window for an Applet

Use the following procedure to add a Deduplication Window for an applet.

### To add a Deduplication Window for an applet

1. Login to Web Tools, open workspace.
2. In the Object Explorer, select the Applet object, and then select the applet of interest, for example, Account Form Applet
3. In the Object Explorer, select Applet User Prop.
4. Add a new record with the following settings:
    - **Name.** DeDuplication Results Applet
    - **Value.** DeDuplication Results (Account) List Applet
5. Save the changes and submit the workspace for delivery.

# Configuring a Real-Time Deduplication Window for Child Applets

Configuration changes are required in Siebel Tools to set up the real-time Deduplication Window on child applets.

To configure the real-time Deduplication Window for a child applet, an applet user property must be added to the respective applet where the Deduplication Window is required. For example, to generate a window from the Account Contact view, add the applet user property to Account Contact List Applet, as described in the following procedure.

## To configure the real-time Deduplication Window for a child applet (Account Contact view)

1. In Web Tools, open the workspace and query for the following applet:

   Account Contact List Applet
2. Add the following user property to this applet:
   - **Name.** DeDuplication Results Applet
   - **Value.** DeDuplication Results (Contact) List Applet
3. Save the changes and submit the workspace for delivery.

# Configuring the Mandatory Fields for Fuzzy Query

For a business component you can configure the mandatory fields for fuzzy query, or query. These fields must include values for the Siebel application to use in Fuzzy Query mode. *Identifying Mandatory Fields for Fuzzy Query* shows the preconfigured mandatory fields that Oracle provides.

Use the following procedure to configure the mandatory fields for a business component.

## To configure the fields that are mandatory for fuzzy query

1. Login to Web Tools, open the workspace.
2. In the Object Explorer, expand Business Component and then select the business component of interest in the Business Components pane.
3. In the Object Explorer, select Business Component User Prop.

   > **Tip:** If the Business Component User Prop object is not visible in the Object Explorer, you can enable it in the Development Tools Options dialog box (View, Options, Object Explorer). If this is necessary, you must repeat Step 2 of this procedure.

4. In the Business Component User Properties pane, select Fuzzy Query Mandatory Fields, and enter the required field names in the Value column.

# Data Quality User Properties

This topic provides information about data quality user properties, including deduplication and data cleansing business service user properties. Information about the following is included in this topic:

- *Account - Business Component User Properties*
- *Contact - Business Component User Properties*
- *List Mgmt Prospective Contact - Business Component User Properties*
- *DeDuplication - Business Service User Properties*
- *Data Cleansing - Business Service User Properties*
- *DQ Sync Services - Business Service User Properties*
- *ISS System Services - Business Services User Properties*

## Account - Business Component User Properties

Data quality uses the Account business component user properties listed in the following table.

| Name | Value | Comment |
|---|---|---|
| DQ Business Object | DQ Vendor | These business component user properties for Account specify the business object and business component used in the data quality Administration views in Siebel CRM. |
| DQ Field Mapping Info BC | DQ Field Mapping Info | |
| DQ Mapping Config BC | DQ Mapping Config | |
| DQ Vendor Info BC | DQ Vendor Info | |
| DQ Key BusComp | Account Key | For the Universal Connector, DQ Key BusComp is used to specify the Name of the buscomp that stored the deduplication key generated in Siebel CRM. |
| DeDuplication Key BusComp | DeDuplication - SSA Account Key | For SDQ Matching Server, DeDuplication Key BusComp is used to specify the Name of the buscomp that stored the dedup key generated by SSA. |
| DeDuplication Results BusComp | DeDuplication Results (Account) | Specifies the Name of the buscomp that will store the returned duplicated record data. |
| DeDuplication Results List Applet | DeDuplication Results (Account) List Applet | Specifies the Name of the pick applet used to prompt the user to resolve duplicates. |
| Fuzzy Query Mandatory Fields | "Name" | Specifies the mandatory fields for Fuzzy Query; that is, the query fields that must include values |

| Name | Value | Comment |
|---|---|---|
| | | so that the Siebel application can use the fuzzy query mode. |
| DQ Associate BC 1 | CUT Address: Address Id | Specifies the Name of the child MVG buscomp, and the field in the parent buscomp that comes from this MVG. This business component applies to the data quality Multiple Address Deduplication feature.<br><br>**Note:** MVG stands for multiple value group. |

For each field used in Multiple Address Deduplication that comes from the child MVG buscomp, a field user property is specified to map it to the child business component field, as shown in the following table.

| Field | Name | Value |
|---|---|---|
| Primary Account City | DQ Associate BC Map | DQ Associate BC 1: City |
| Primary Account Country | DQ Associate BC Map | DQ Associate BC 1: Country |
| Primary Account Postal Code | DQ Associate BC Map | DQ Associate BC 1: Postal Code |
| Primary Account State | DQ Associate BC Map | DQ Associate BC 1: State |
| Primary Account Street Address | DQ Associate BC Map | DQ Associate BC 1: Street Address |

# Contact - Business Component User Properties

Data quality uses the Contact business component user properties listed in the following table.

| Name | Value | Comment |
|---|---|---|
| DQ Business Object | DQ Vendor | These business component user properties for Contact specify the business object and business component used in the data quality Administration views in Siebel CRM. |
| DQ Field Mapping Info BC | DQ Field Mapping Info | |
| DQ Mapping Config BC | DQ Mapping Config | |
| DQ Vendor Info BC | DQ Vendor Info | |

| Name | Value | Comment |
|---|---|---|
| DQ Key BusComp | Contact Key | For the Universal Connector, DQ Key BusComp is used to specify the Name of the buscomp that stored the deduplication key generated in Siebel CRM. |
| DeDuplication Key BusComp | DeDuplication - SSA Contact Key | For SDQ Matching Server, DeDuplication Key BusComp is used to specify the Name of the buscomp that stored the dedup key generated by SSA. |
| DeDuplication Results BusComp | DeDuplication Results (Contact) | Specifies the Name of the buscomp that will store the returned duplicated record data. |
| DeDuplication Results List Applet | DeDuplication Results (Contact) List Applet | Specifies the Name of the pick applet used to prompt the user to resolve duplicates. |
| Fuzzy Query Mandatory Fields | "Last Name", "First Name" | Specifies the mandatory fields for Fuzzy Query; that is, the query fields that must include values so that the Siebel application can use the fuzzy query mode. |
| DQ Associate BC 1 | CUT Address: Personal Address Id | Specifies the Name of the child MVG buscomp, and the field in the parent buscomp that comes from this MVG. This business component applies to the data quality Multiple Address Deduplication feature.<br><br>**Note:** MVG stands for multiple value group. |

For each field used in Multiple Address Deduplication that comes from the child MVG buscomp, a field user property is specified to map it to the child business component field, as shown in the following table.

| Field | Name | Value |
|---|---|---|
| Primary Personal City | DQ Associate BC Map | DQ Associate BC 1: City |
| Primary Personal Country | DQ Associate BC Map | DQ Associate BC 1: Country |
| Primary Personal Postal Code | DQ Associate BC Map | DQ Associate BC 1: Postal Code |
| Primary Personal State | DQ Associate BC Map | DQ Associate BC 1: State |
| Primary Personal Street Address | DQ Associate BC Map | DQ Associate BC 1: Street Address |

# List Mgmt Prospective Contact - Business Component User Properties

Data quality uses the List Mgmt Prospective Contact business component user properties listed in the following table.

| Name | Value | Comment |
|---|---|---|
| DQ Business Object | DQ Vendor | These business component user properties for List Mgmt Prospective Contact specify the business object and business component used in the data quality Administration views in Siebel CRM. |
| DQ Field Mapping Info BC | DQ Field Mapping Info | |
| DQ Mapping Config BC | DQ Mapping Config | |
| DQ Vendor Info BC | DQ Vendor Info | |
| DQ Key BusComp | Prospect Key | For the Universal Connector, DQ Key BusComp is used to specify the Name of the buscomp that stored the deduplication key generated in Siebel CRM. |
| DeDuplication Key BusComp | DeDuplication - SSA Prospect Key | For SDQ Matching Server, DQ Key BusComp is used to specify the Name of the buscomp that stored the dedup key generated by SSA. |
| DeDuplication Results BusComp | DeDuplication Results (Prospect) | Specifies the Name of the buscomp that will store the returned duplicated record data. |
| DeDuplication Results List Applet | DeDuplication Results (Prospect) List Applet | Specifies the Name of the pick applet used to prompt the user to resolve duplicates. |
| Fuzzy Query Mandatory Fields | "Last Name", "First Name" | Specifies the mandatory fields for Fuzzy Query; that is, the query fields that must include values so that the Siebel application can use the fuzzy query mode. |

# DeDuplication - Business Service User Properties

Data quality uses the DeDuplication business service user properties listed in the following table.

| Name | Value | Comment |
|---|---|---|
| DQ Vendor BO | DQ Vendor | These business service user properties for DeDuplication specify the business object and business component used in the data quality Administration views in Siebel CRM. |
| DQ Vendor BC | DQ Vendor Info | |
| DQ Mapping Config BC | DQ Mapping Config | |
| DQ Field Mapping Info BC | DQ Field Mapping Info | |
| DQ Vendor Param BC | DQ Vendor Parameter | |

# Data Cleansing - Business Service User Properties

Data quality uses the Data Cleansing business service user properties listed in the following table.

| Name | Value | Comment |
|---|---|---|
| DQ Vendor BO | DQ Vendor | These business service user properties for Data Cleansing specify the business object and business component used in the data quality Administration views in Siebel CRM. |
| DQ Vendor BC | DQ Vendor Info | |
| DQ Mapping Config BC | DQ Mapping Config | |
| DQ Field Mapping Info BC | DQ Field Mapping Info | |
| DQ Vendor Param BC | DQ Vendor Parameter | |

# DQ Sync Services - Business Service User Properties

The DQ Sync Services business service is used when synchronizing data between the Siebel application and Oracle Data Quality Matching Server (Informatica Identity Resolution). The Siebel application uses it to insert records into the NSA table belonging to the Oracle Data Quality Matching Server.

Data quality uses the DQ Sync Services business service user properties listed in the following table.

| Name | Value | Comment |
|---|---|---|
| Account Integration Object Name | SyncAccount | These business service user properties specify the Integration Objects and Components |

| Name | Value | Comment |
|---|---|---|
| Account Address Component Name | Account_Business Address | used by the Siebel application to synchronize with Oracle Data Quality Matching Server (Informatica Identity Resolution). |
| Contact Integration Object Name | SyncContact | |
| Contact Address Component Name | Contact_INS Personal Address | |
| Prospect Integration Object Name | SyncProspect | |
| ISS System Name | siebeldq | These business service user properties specify the default Informatica Identity Resolution system name and IDT table names. |
| Account IDT Name | IDS_01_IDT_ACCOUNT | |
| Contact IDT Name | IDS_01_IDT_CONTACT | |
| Prospect IDT Name | IDS_01_IDT_PROSPECT | |
| Account Record Length | 1090 | These business service user properties specify the Account record fields, data type, and length. |
| Account_DeDupFlds | Name\|DUNS Number\|Id | |
| Account_DataType | W\|W\|C | |
| Account_ExtLen | 200\|120\|30 | |
| Contact Record Length | 1850 | These business service user properties specify the Contact record fields, data type, and length. |
| Contact_DeDupFlds | Birth Date\|Cellular Phone #\|Email Address\|First Name Last Name\|Home Phone #\|Middle Name\|Account\|Id\|Social Security Number\|Work Phone # | |
| Contact_DataType | W\|W\|W\|W\|W\|W\|W\|C\|W\|W | |
| Contact_ExtLen | 120\|120\|120\|200\|120\|200\|200\|30\|120\|120 | |
| Prospect Record Length | 1770 | These business service user properties specify the Prospect record fields, data type, and length. |
| Prospect_DeDupFlds | Account\|Cellular Phone #\|City\|Country\|Email Address\|First Name Last Name\|Home Phone #\|Middle Name\|Postal Code\|Social Security Number\|State\|Street Address\|Work Phone #\|Id | |
| Prospect_DataType | W\|W\|W\|W\|W\|W\|W\|W\|W\|W\|W\|W\|C | |

| Name | Value | Comment |
|---|---|---|
| Prospect_ExtLen | 200\|120\|120\|60\|120\|200\|120\|200\|40\|120\|40\|200\|200\|30 | |
| Account_Business Address_DeDupFlds | City\|Country\|Postal Code\|State\|Street Address\|Address Id | These business service user properties specify the Account Business Address record fields, data type, and length. |
| Account_Business Address_DataType | W\|W\|W\|W\|W\|C | |
| Account_Business Address_ExtLen | 200\|120\|120\|40\|200\|60 | |
| Contact_INS Personal Address_DeDupFlds | INS Personal City\|INS Personal Country\|INS Personal Postal Code\|INS Personal State\|INS Personal Street Address\|INS Personal Address Id | These business service user properties specify the Contact INS Personal Address record fields, data type, and length. |
| Contact_INS Personal Address_DataType | W\|W\|W\|W\|W\|C | |
| Contact_INS Personal Address_ExtLen | 120\|40\|40\|40\|200\|60 | |
| Filter Characters | *<Enter characters separated by a single space>*<br><br>**Note:** The following special characters are removed automatically from all data (so do not specify them in the DQ Sync Services Business Service user property):<br>^ (code separator)<br>\| (code separator)<br>\r (carriage return)<br>\n (new line)<br>\t (horizontal tab)<br>\f (form feed) | This business service user property is used to specify any special characters that need to be removed from data sent to the Oracle Data Quality Matching Server.<br><br>**Note:** Enter all filter characters separated by a single space. For control characters, enter the ASCII code in the following format: **^nnn** (where **nnn** is the code in decimal). |
| IIR Server on Little Endian Operating System | Yes | This business service user property is used to specify the *Endian* of the Operating System where the Oracle Data Quality Matching Server is installed. |

## ISS System Services - Business Services User Properties

When synchronizing data in Multiple Languages Support mode between the Siebel application and Oracle Data Quality Matching Server (Informatica Identity Resolution), the ISS System Services Business Service is used to specify the system name and database number in Informatica Identity Resolution, where each system corresponds to a specific country or language.

Siebel Data quality uses the ISS System Services business service user properties listed in the following table.

| Name | Value | Comment |
|---|---|---|
| Account Default System | siebeldq | These business service user properties specify the default Informatica Identity Resolution system name for each support object. |
| Contact Default System | siebeldq | |
| Prospect Default System | siebeldq | |
| siebeldq | 01 | These are sample business service user properties. You must enter the actual Informatica Identity Resolution system name and corresponding database number. The default system has a database number of 01. |
| siebeldq_Denmark | 02 | |
| siebeldq_UNITED_STATES | 03 | |

### Related Topic

*Configuring Multiple Language Support for Data Matching*

# Optimizing Data Cleansing Performance

The following are recommendations for achieving good performance with data cleansing when working with large volumes of data:

- Include only new or recently modified records in the batch data cleansing process.
- Cleansing all records in the Siebel database each time a data cleansing is performed can cause performance issues. Include an Object WHERE clause when you submit your batch job, as shown in the following table. Split the tasks into smaller tasks and run them concurrently.

| To Cleanse | Use This in Your Object WHERE Clause |
|---|---|
| Updated records | [Last Clnse Date] < [Updated] |
| New records | [Last Clnse Date] IS NULL |

| To Cleanse | Use This in Your Object WHERE Clause |
|---|---|
| | |
| Updated and new records | [Last Clnse Date] < [Updated] OR [Last Clnse Date] IS NULL |

To speed up the data cleansing task for large databases, run batch jobs to cleanse a smaller number of records at a time using an Object WHERE clause. For more information about data cleansing for large batches, see *Cleansing Data Using Batch Jobs*.

# Optimizing Data Matching Performance

The following are recommendations for achieving good performance with data matching when working with large volumes of data:

- Work with a database administrator to verify that the table space is large enough to hold the records generated during the data matching process.

  During the batch data matching process, the information on potential duplicate records is stored in the S_DEDUP_RESULT table as a pair of row IDs of the duplicate records and the match scores between them. The number of records in the results table S_DEDUP_RESULT can include up to six times the number of records in the base tables combined. Remember that:

  - If the base tables contain many duplicates, more records are inserted in the results table.
  - If different search types are used, a different set of duplicate records might be found and will be inserted into the results table.
  - If you use a low match threshold, the matching process generates more records to the results table.

- Remove obsolete result records manually from the S_DEDUP_RESULT table by running SQL statements directly on this table.

  When a duplicate record is detected, the information about the duplicate is automatically placed in the S_DEDUP_RESULT table, whether or not the same information exists in that table. Running multiple batch data matching tasks therefore results in a large number of duplicate records in the table. Therefore, it is recommended that you manually remove the existing records in the S_DEDUP_RESULT table before running a new batch data matching task. You can remove the records using any utility that allows you to submit SQL statements.

  > **Note:** When truncating the S_DEDUP_RESULT table, all potential duplicate records found for all data matching business components are deleted.

- Match performance is slow when the SOH string is set as follows:

  ```
  S_POSTN_CON.CON_LAST_NAME = 'SOH'
  ```

  Setting the following values improves match performance:

  - Navigate to the Contact Business Component.
  - Add the following user property and value under the Last Name field:

| User Property | Value |
| --- | --- |
| Sort Search Optimization | FALSE |

For more information about running batch data matching, see *Matching Data Using Batch Jobs*.

# 6 Configuring Data Quality with Oracle Enterprise Data Quality

## Configuring Data Quality with Oracle Enterprise Data Quality

This chapter describes the configuration tasks that you can perform for Oracle Enterprise Data Quality. It covers the following topics:


- *Overview of Oracle Enterprise Data Quality*
- *Installing Oracle Enterprise Data Quality (EDQ)*
- *Process of Configuring Oracle Enterprise Data Quality for Matching and Address Verification*
- *Process of Configuring the Staging Database for Oracle Enterprise Data Quality*
- *Process of Configuring Oracle Enterprise Data Quality to Find Contact and Address Matches*
- *Vendor Parameters for Oracle Enterprise Data Quality*
- *Oracle Enterprise Data Quality Field Mappings for Siebel Business Components*
- *Configuring Third-Party Key Generation for Oracle Enterprise Data Quality*
- *Enabling Support of Multiple Child Entity Deduplication for Oracle Enterprise Data Quality*
- *Data Format for Deduplication of Multiple Child Entities*
- *Overview of Configuring Oracle Enterprise Data Quality for Deduplication of Child Entities*
- *Process of Configuring Additional Child Entities for Deduplication*
- *Sample Message from Siebel CRM to EDQ Siebel Connector for Additional Child Entity Deduplication*
- *Overview of Configuring Social Media Child Objects for Matching*
- *Sample Message from Siebel CRM to EDQ Siebel Connector for Social Media Child Object Matching*


## Overview of Oracle Enterprise Data Quality

The Oracle Enterprise Data Quality suite of products helps organizations by delivering fit-for-purpose data. Oracle Enterprise Data Quality products enable individuals and collaborative teams to identify and resolve problems in underlying data. With Oracle Enterprise Data Quality products, you can identify new opportunities, improve operational efficiency, and more effectively comply with industry or governmental regulation.

The most common data domains in data quality are customer (or more generally, party data which includes suppliers, employees and so on) and product data. Oracle Enterprise Data Quality products address these differences by providing you with the capability to address each.

# Installing Oracle Enterprise Data Quality (EDQ)

To install Oracle Enterprise Data Quality, see *http://www.oracle.com/webfolder/technetwork/data-quality/edqdocs/oedq_install.pdf*.

> **Note:** To support the Oracle Enterprise Data Quality Address Verification with Siebel 8.1.1.11 FP, you must install Oracle Enterprise Data Quality Customer Data Services Pack (EDQ-CDS) version 9.0.4 or later.

## Requirements for Installing Oracle Enterprise Data Quality

The following requirements pertain to installation Oracle Enterprise Data Quality (EDQ). For installation instructions, see *http://www.oracle.com/technetwork/middleware/oedq/documentation/index.html*.

- Oracle Enterprise Data Quality (EDQ) Server following the Oracle Enterprise Data Quality (EDQ) Installation Notes from Datanomic.
- Oracle Enterprise Data Quality Customer Data Services Pack (EDQ-CDS) must be installed following the instruction from Oracle Enterprise Data Quality Installation Guide.
- You must initialize reference data after installing Oracle Enterprise Data Quality Customer Data Services Pack (EDQ-CDS).
- You must install Oracle Enterprise Data Quality Address Verification (EDQ-AV) in accordance with instructions with its installation guide. This is required if you want to run the Address Clean process with Oracle Enterprise Data Quality Address Verification (EDQ-AV).

  > **Note:** Oracle Enterprise Data Quality Address Verification (EDQ-AV) requires the Global Knowledge Repository, which comprises the data used by Address Verification. This component must be purchased directly from Oracle's partner Loqate (*http://www.loqate.com/)*. The repository includes its own installer for Windows and other platforms.

- Oracle Enterprise Data Quality-Siebel Connector must be installed.
- The Java Runtime Environment (JRE) version 6 or later must be installed on the Siebel server.
- The version of JRE deployed must match the architecture of the server supported by Siebel CRM. For example, as Siebel supports 32-bit versions of Windows, the 32-bit version of JRE must be installed.

## Compatibility of Released Versions of Oracle Enterprise Data Quality Customer Data Services Pack (EDQ-CDS)

The following table lists version compatibility information for Oracle Enterprise Data Quality Customer Data Services Pack (EDQ-CDS).

| Oracle Enterprise Data Quality Customer Data Services Pack | Oracle Enterprise Data Quality | Siebel Connector for Oracle Enterprise Data Quality | EDQ Address Verfication (EDQ-AV) |
|---|---|---|---|
| 9.0.1 | 12.2.1.1.0 | 9.0.3 to 9.0.5 | Any |

| Oracle Enterprise Data Quality Customer Data Services Pack | Oracle Enterprise Data Quality | Siebel Connector for Oracle Enterprise Data Quality | EDQ Address Verfication (EDQ-AV) |
|---|---|---|---|
| | | | |
| 9.0.2 | 12.2.1.1.0 | 9.0.4 to 9.0.5 | Any |
| 9.0.3 | 12.2.1.1.0 | 9.0.4 to 9.0.5 | Any |
| 9.0.4 | 12.2.1.1.0 | 9.0.6 | 12.4.0.0.0 or later |

# Process of Configuring Oracle Enterprise Data Quality for Matching and Address Verification

Siebel Connector calls Oracle Enterprise Data Quality jobs and Web services for real-time data cleansing and matching. A shared staging database is used to pass data between the Siebel CRM application, and Oracle Enterprise Data Quality when batch jobs are run.

This process includes the following tasks:

- *Configuring the Siebel Server Under Windows*
- *Configuring the Siebel Server Under UNIX*
- *Configuring the Siebel Server to Use Oracle Enterprise Data Quality*
- *Configuring a New Data Quality Vendor*
- *Disabling Informatica Identity Resolution Synchronization*

## Configuring the Siebel Server Under Windows

This task is a step in *Process of Configuring Oracle Enterprise Data Quality for Matching and Address Verification*.

### To configure Siebel Server under Windows

1. Copy the dnd.dll file to the `Siebel_Server_root\bin` directory.

   This file is located in the Native sub-folder where you installed the EDQ-Siebel Connector.
2. Create a file in the `Siebel_Server_root\SDQConnector\` directory with the name *dnd.parms*. This file is used to point to the installation directory on the same server where the Siebel connector runs. The file must contain the following lines:

   ```
   javalib: [location of the JRE]/bin/client/jvm.dll

   directory: [the installation directory for the connector]
   ```

   For example:

   ```
   javalib: C:/Program Files/Java/jre1.6/bin/client/jvm.dll
   ```

```
directory: C:/SiebelConnector
```

> **Note:** The Siebel connector is not language specific, so is not installed in a specific language directory.

3. If detailed trace information on connector requests is required, add the following lines to the dnd.parms file to enable logging:

```
logfile: sdq%05d.log

jlogfile: jsdq%05d.log
```

4. Extract the siebelconnector.zip file into the installation directory specified in the dnd.parms file (the default location is: `Siebel_Server_root\dnd\install`.) This installs the following files:
   - Jar files for the connector.
   - Library jar files needed by the connector.
   - Properties files used to store configuration information for the Siebel Connector.
5. Extract and copy the dnd.properties file from the `\oEDQ Customer Data Services Pack 9.0.3\edq-cds-9_0_3_(330)\properties\dnd` properties file into the EDQ-Siebel Connector root directory specified in the dnd.params file directory parameter.

# Configuring the Siebel Server Under UNIX

This task is a step in *Process of Configuring Oracle Enterprise Data Quality for Matching and Address Verification*.

## To configure AIX, Linux and Solaris Siebel Server

1. Copy the libdnd.so driver file to the `Siebel_Server_root/lib` directory.

   The driver files for each operating system are held in the native subfolder of the siebelconnector.zip file in the Oracle Enterprise Data Quality installation:
   - AIX: `/native/aix/ppc/` or `/native/ppc64`, depending on whether a 32-bit or 64-bit version of AIX is in use.
   - Linux: `/native/linux`
   - Solaris: `/native/sparcv9`
2. Create a file in the `Siebel_Server_root/SDQConnector` directory with the name dnd.parms. This file will be used to point to the installation directory on the same server where the Siebel connector will run. The file must contain the following lines:

```
javalib: [location of the JRE]/lib/i386/client/libjvm.so

directory: [the installation directory for the connector]
```

   For example:

```
javalib: /usr/java/jre1.6/lib/i386/client/libjvm.so

directory: /opt/siebel/dnd/install
```

3. If detailed trace information on connector requests is required, add the following lines to the dnd.parms file to enable logging:

```
logfile: sdq%05d.log

jlogfile: jsdq%05d.log
```

Extract the siebelconnector.zip file into the installation directory specified in the dnd.parms file (the default being /opt/siebel/dnd/install). This installs the following files:

- Jar files for the connector
- Library jar files needed by the connector
- Properties files used to store configuration information for the Siebel Connector

4. Extract and copy the dnd.properties file in the EDQ-CDS.zip file over the identically named file in the installation directory.

# Configuring the Siebel Server to Use Oracle Enterprise Data Quality

This task is a step in *Process of Configuring Oracle Enterprise Data Quality for Matching and Address Verification*.

## To configure the Siebel server to use Oracle Enterprise Data Quality

1. Log into the Siebel web client as a system administrator.
2. Navigate to Administration - Server Configuration.
3. Click the Enterprises link and select the Component Groups tab.
4. Locate the Data Quality component group and make sure it is enabled.
5. In the Component Group Assignments view, make sure the component group has been assigned and enabled on the Siebel server.
6. Click the Parameters tab, and in the Enterprise Parameters list, locate Data Cleaning Type, and set the value to Data Cleansing Type.
7. Locate the DeDuplication Data Type parameter, and set the value to EDQ.
8. Click the Servers link, and locate the Data Quality Manager component, then click the Parameters tab.
9. Make sure the following parameter values are set to EDQ:
   - Data Cleansing Type
   - DeDuplication Data Type
10. In the Component Parameters view, make sure the following parameters are set to True:
    - Data Cleansing Enable Flag
    - DeDuplication Enable Flag
11. Find the component for the Siebel application which is to use EDQ-CDS, such as the UCM Object Manager (ENU).
12. Repeat Step 5 through Step 11 for any other components that you want to enable for EDQ-CDS.

# Configuring a New Data Quality Vendor

This task is a step in *Process of Configuring Oracle Enterprise Data Quality for Matching and Address Verification*.

## To configure a new data quality vendor

1. Log into the Siebel Web client as a system administrator.
2. Navigate to Administration - Data Quality.
3. Click Data Quality Settings and enable the following options:
   - Enable DataCleansing = Yes
   - Enable DeDuplication = Yes
4. Add the following options to enable a pop-up dialog box to display possible duplicate records for each new record of a given type when the real-time duplicate prevention service is running.

   If these options are not specified, and the real-time duplicate prevention service is running, the check will still be performed, but you must navigate to the Duplicate Accounts, Duplicate Contacts or Duplicate Prospects tab when adding a new record:
   - Force User DeDupe - Account = Yes
   - Force User DeDupe - Contact = Yes
   - Force User DeDupe - List Mgmt = Yes
5. Click the Third Party Administration link, and in the Vendor list add a new entry with EDQ as the Name value, and dnd at the DLL Name value.
6. With the newly created vendor record selected, click the BC Vendor Field Mapping tab the enter the following information:

| Business Component Name | Operation |
|---|---|
| Account | Data Cleansing |
| Account | DeDuplication |
| Contact | Data Cleansing |
| Contact | DeDuplication |
| CUT Address | Data Cleansing |
| List Mgmt Prospective Contact | Data Cleansing |
| List Mgmt Prospective Contact | DeDuplication |
| Personal Address | Data Cleansing |

7. If multiple child entities are enabled, add the following entries:

| Business Component Name | Operation |
|---|---|
| CUT Address | DeDuplication |
| Personal Address | DeDuplication |

8. In the Field mappings list, set the field mappings for each business component you entered in the BC Operation list. For more information, see *Mapping of Vendor Fields to Business Component Fields*.

   **Note:** Field mappings can be added or changed as required, provided that they correspond to the configured Web Services in Oracle Enterprise Data Quality (for real-time operations), and the names of the columns in the tables of the staging database (for batch operations).

9. In the Vendor screen, make sure the EDQ vendor is still selected, then click the Vendor Parameter tab.
10. Add the parameters listed in *Configuring Vendor Parameters*.

## Disabling Informatica Identity Resolution Synchronization

This task is a step in *Process of Configuring Oracle Enterprise Data Quality for Matching and Address Verification*.

**Note:** This task is required only for users of Oracle Data Quality Matching Server, which uses Informatica Identity Resolution software. This task allows you to disable synchronization used by Informatica Identity Resolution.

### To disable Informatica Identity Resolution synchronization

1. Navigate to Administration - Runtime Events, then Action Sets.
2. In the Action Sets view, query for DQ Sync in the Action Sets field.
3. Deselect the Active flag to inactivate all related action sets.
4. Navigate to Administration - Data Quality, then Data Quality Settings.
5. In the Data Quality Settings view, set the Enable DQ Sync parameter value to No.

# Process of Configuring the Staging Database for Oracle Enterprise Data Quality

The staging database is used by the Siebel Connector as a staging area for batch matching jobs on the primary data.

**Note:** In Oracle Customer Hub (UCM), batch matching is normally performed using the Oracle Enterprise Data Quality real-time matching services, so that all connected source systems are considered during matching. However, for both Siebel CRM and Oracle Customer Hub (UCM) the Siebel Data Quality Manager service can run batch jobs on the primary data only.

Candidates for matching are written to a table to be read by an Oracle Enterprise Data Quality job. Oracle Enterprise Data Quality then exports duplicates from the matching process to a table to be read back into the Siebel database.

This process includes the following tasks:

- *Creating Tables*
- *Configuring Connections*

## Creating Tables

This task is a step in *Process of Configuring the Staging Database for Oracle Enterprise Data Quality*.

The sql folder in the root of the EDQ-CDS distribution contains six scripts that can be run to create tables as follows:

- siebel_account_candidates.sql creates the SIEBEL_ACCOUNT_CANDIDATES table.
- siebel_account_matches.sql creates the SIEBEL_ACCOUNT_MATCHES table.
- siebel_contact_candidates.sql creates the SIEBEL_CONTACT_CANDIDATES table.
- siebel_contact_matches.sql creates the SIEBEL_CONTACT_MATCHES table.
- siebel_prospect_candidates.sql creates the SIEBEL_PROSPECT_CANDIDATES table.
- siebel_prospect_matches.sql creates the SIEBEL_PROSPECT_MATCHES table.

## Configuring Connections

This task is a step in *Process of Configuring the Staging Database for Oracle Enterprise Data Quality*.

Both the Siebel Connector and EDQ-CDS itself must connect to the staging database to read and write to the Candidate and Match tables when processing batch jobs. These tables can be created in any database and schema, but by default the connection is configured for an Oracle database.

The connection details are specified in the dnd.properties file, see *Process of Configuring Oracle Enterprise Data Quality for Matching and Address Verification*. To configure these, open the dnd.properties file and edit the six parameters in the Staging database connection details section. These parameters control the database host, port, credentials and other settings used to connect to the staging database.

# Process of Configuring Oracle Enterprise Data Quality to Find Contact and Address Matches

By default, Oracle Enterprise Data Quality does not find matches with the following search constraints: Contact First and Last Name only or Account Name and Site only. A match is made only when addresses are supplied for the contact or account. To find matches with Contact First and Last Name only or Account Name and Site only without addresses, you must activate a rule within Oracle Enterprise Data Quality.

To configure Oracle Enterprise Data Quality, perform the following tasks:

1. *Activating Oracle Enterprise Data Quality Rules*
2. *Configuring Siebel Match Threshold Settings*

# Activating Oracle Enterprise Data Quality Rules

By default, Oracle Enterprise Data Quality does not find matches with the following search constraints: Contact First and Last Name only or Account Name and Site only. A match is only made when addresses are supplied for the contact or account. To find matches with Contact First and Last Name only or Account Name and Site only without addresses, you must activate a rule within Oracle Enterprise Data Quality.

This task is a step in *Process of Configuring Oracle Enterprise Data Quality to Find Contact and Address Matches*.

## To activate Oracle Enterprise Data Quality rules

1. In Oracle Enterprise Data Quality, navigate to Director, then open the following project: EDQ-CDS.
2. Open the following processes:
   - Individual Match - Real-Time
   - Individual Match - Batch
3. For each process, open the Match Individuals step, then click the Match icon to open the match rules.
4. Select the checkbox to activate the following rules:
   - [I010U] Script full name exact; no address. Set the value to 65.
   - [I020U] Name exact; no address. Set the value to 65.
5. Repeat each Step 2 through Step 4 for the following processes:
   - Account Entity Match - Real-Time
   - Entity Match - Batch

   The rule returns a score of only 60 there is a match on Account Name and Site fields.
6. Save the project.
7. Restart Oracle Enterprise Data Quality services and the Real-time START ALL jobs.

# Configuring Siebel Match Threshold Settings

Because the Oracle Enterprise Data Quality rule now has matching scores of 60 and 65, you must change the Siebel application Match Threshold setting so that your Siebel application can identify the new matches.

This task is a step in *Process of Configuring Oracle Enterprise Data Quality to Find Contact and Address Matches*.

## To change the Siebel match threshold setting

1. Navigate to Administration - Data Quality, then Data Quality Settings.
2. Change the Match Threshold setting from the default value of 70 to 50.
3. Restart Siebel Server services.

# Vendor Parameters for Oracle Enterprise Data Quality

The following table lists parameters which allow you to enable and control third party data quality vendors.

| Name | Value |
|---|---|
| Key Generation Process by Third Party | The default value is No.<br><br>This parameter should only be set to Yes if external key generation is supported by the installed version of Siebel. For more information, see *Oracle Customer Hub (UCM) Master Data Management Reference* .<br><br>If this parameter is set to No, the Query Expression parameters and the Token Expression parameters must also be set. For more information, see *Match Key Generation Using the Universal Connector with Third-Party Data Quality Vendors*. |
| Support Multiple Child Entities Deduplication | The default value is Yes.<br><br>**Note:** If this parameter is set to Yes, the CUT and Personal Address DeDup Record Type parameters, which are displayed in this table, are enabled. Also note that this parameter should only be set to Yes if Multiple Child Entities are supported by the installed version of Siebel CRM. |
| Account DataCleanse Record Type | accountclean. |
| Account DeDup Record Type | accountmatch. |
| CUT Address DataCleanse Record Type | addressclean. |
| CUT Address DeDup Record Type | addressmatch.<br><br>**Note:** Only used if multiple child entities are enabled. |
| Personal Address DataCleanse Record Type | addressclean. |
| Personal Address DeDup Record Type | addressmatch.<br><br>**Note:** Only used if multiple child entities are enabled. |
| Contact DataCleanse Record Type | contactclean. |
| Contact DeDup Record Type | contactmatch. |
| List Mgmt Prospective Contact DataCleanse Record Type | prospectclean. |
| List Mgmt Prospective Contact DeDup Record Type | prospectmatch. |
| Batch Max Num of Records | 200 |
| Parameter 1 | "session", "clusterlevel", "3". |

| Name | Value |
|---|---|
| | **Note:** There must be a space after each comma. |
| Parameter 2 | "session", "matchthreshold", "50".<br>**Note:** There must be a space after each comma. |

# Oracle Enterprise Data Quality Field Mappings for Siebel Business Components

The following field mappings work with the provided templates for data quality services, and the provided example SQL for creating the staging database.

Note the following:

- All fields in these business components are drop-down fields, with the exception of Id. This field must be completed manually.
- If Oracle Enterprise Data Quality Address Verification is used for address cleaning:
  - The State field mapping only functions correctly if all addresses in the Siebel database are US addresses. If the addresses are not all US addresses, do not map this field.
  - The Country field mapping may not function correctly if the Siebel CRM field is restricted to fixed list of values that do not exactly match those returned by the cleaning process.

This topic contains information on the following field mappings:

- *Account - Data Cleansing*
- *Account - DeDuplication*
- *CUT Address - Data Cleansing*
- *CUT Address - DeDuplication*
- *Personal Address - Data Cleansing*
- *Personal Address - DeDuplication*
- *Contact - Data Cleansing*
- *Contact - DeDuplication*
- *List Mgmt Prospective Contact - Data Cleansing*
- *List Mgmt Prospective Contact - DeDuplication*

## Account - Data Cleansing

The following table lists the mapped fields for data cleaning of the Account business component.

| Business Component Field | Mapped Field |
|---|---|
| Id | entityid |
| Language Code | languages |
| Name | name |
| Location | subname |
| Main Phone Number | phone |
| Tax ID Number | taxnumber |
| Home Page | website |
| VAT registration number | vatnumber |

# Account - DeDuplication

The following table lists the mapped fields for deduplication of the Account business component.

| Business Component Field | Mapped Field |
|---|---|
| Id | entityid |
| Language Code | languages |
| Name | name |
| Location | subname |
| Main Phone Number | phone |
| Tax ID Number | taxnumber |
| VAT registration number | vatnumber |
| Home Page | website |

The following table lists the mapped fields for deduplication of the Account business component when the Multiple Child Entities parameter is disabled.

| Business Component Field | Mapped Field |
|---|---|
| Primary Account Street Address | address1 |
| Primary Account Address Street Address2 | address2 |
| Primary Account Address Street Address3 | address3 |
| Primary Account City | city |
| Primary Account State | adminarea |
| Primary Account Postal Code | postalcode |
| Primary Account Country | country |

# CUT Address - Data Cleansing

The following table lists the mapped fields for data cleansing of the CUT Address business component.

| Business Component Field | Mapped Field |
|---|---|
| Id | addressid |
| Street Address | address1 |
| Street Address 2 | address2 |
| Street Address 3 | address3 |
| Street Address 4 | address4 |
| City | city |
| Neighborhood | doubledependentlocality |
| Metro Area | dependentlocality |

| Business Component Field | Mapped Field |
|---|---|
| County | subadminarea |
| State | adminarea |
| Postal Code | postalcode |
| Country | country |

# CUT Address - DeDuplication

The following table lists the mapped fields for deduplication of the CUT Address business component.

**Note:** This business component is only used if the Multiple Child Entities parameter is enabled.

| Business Component Field | Mapped Field |
|---|---|
| Street Address | address1 |
| Street Address 2 | address2 |
| City | city |
| State | adminarea |
| Postal Code | postalcode |
| Country | country |

# Personal Address - Data Cleansing

The following table lists the mapped fields for data cleansing of the Personal Address business component.

| Business Component Field | Mapped Field |
|---|---|
| Id | addressid |
| Street Address | address1 |

| Business Component Field | Mapped Field |
|---|---|
| | |
| Street Address 2 | address2 |
| Street Address 3 | address3 |
| Street Address 4 | address4 |
| City | city |
| Neighborhood | doubledependentlocality |
| Metro Area | dependentlocality |
| County | subadminarea |
| State | adminarea |
| Postal Code | postalcode |
| Country | country |

# Personal Address - DeDuplication

The following table lists the mapped fields for deduplication of the Personal Address business component.

**Note:** This business component is only used if the Multiple Child Entities parameter is enabled.

| Business Component Field | Mapped Field |
|---|---|
| Street Address | address1 |
| Street Address 2 | address2 |
| City | city |
| State | adminarea |
| Postal Code | postalcode |

| Business Component Field | Mapped Field |
|---|---|
| Country | country |

# Contact - Data Cleansing

The following table lists the mapped fields for data cleansing the Contact business component.

| Business Component Field | Mapped Field |
|---|---|
| Id | individualid |
| Language Code | languages |
| M/M | title |
| M/F | gender |
| First Name | firstname |
| Middle Name | middlename |
| Last Name | lastname |
| Home Phone # | homephone |
| Work Phone # | workphone |
| Fax Phone # | faxphone |
| Cellular Phone # | mobilephone |
| Email Address | email |
| Job Title | jobtitle |
| Social Security Number | taxnumber |
| Birth Date | dob |

# Contact - DeDuplication

The following table lists the mapped fields for deduplication of the Contact business component.

| Business Component Field | Mapped Field |
|---|---|
| Id | individualid |
| Language Code | languages |
| M/M | title |
| M/F | gender |
| First Name | firstname |
| Middle Name | middlename |
| Last Name | lastname |
| Home Phone # | homephone |
| Work Phone # | workphone |
| Fax Phone # | faxphone |
| Cellular Phone # | mobilephone |
| Email Address | email |
| Job Title | jobtitle |
| Social Security Number | taxnumber |
| Birth Date | dob |
| Primary Account Name | accountname |

The following table lists the mapped fields for deduplication of the Contact business component when the Multiple Child Entities parameter is disabled.

| Business Component Field | Mapped Field |
|---|---|
| Primary Street Address | address1 |
| Primary Address Street Address2 | address2 |
| Primary Address Street Address3 | address3 |
| Primary City | city |
| Primary State | adminarea |
| Primary Postal Code | postalcode |
| Primary Country | country |

# List Mgmt Prospective Contact - Data Cleansing

The following table lists the mapped fields for data cleansing the List Mgmt Prospective Contact business component.

| Business Component Field | Mapped Field |
|---|---|
| Id | individualid |
| M/M | title |
| M/F | gender |
| First Name | firstname |
| Middle Name | middlename |
| Last Name | lastname |
| Home Phone # | homephone |
| Work Phone # | workphone |
| Fax Phone # | faxphone |
| Cellular Phone # | mobilephone |

| Business Component Field | Mapped Field |
|---|---|
| | |
| Email Address | email |
| Job Title | jobtitle |
| Social Security Number | taxnumber |
| Birth Date | dob |

# List Mgmt Prospective Contact - DeDuplication

The following table lists the mapped address fields for deduplication of the List Mgmt Prospective Contact business component.

**Note:** List Mgmt Prospective Contact has just one address. You can, however, include the address fields in the mapped fields for deduplication of List Mgmt Prospective Contact, regardless of whether the Multiple Child Entities parameter are enabled or not.

| Business Component Field | Mapped Field |
|---|---|
| Id | individualid |
| M/M | title |
| M/F | gender |
| First Name | firstname |
| Middle Name | middlename |
| Last Name | lastname |
| City | city |
| Country | country |
| Postal Code | postalcode |
| State | adminarea |

| Business Component Field | Mapped Field |
|---|---|
| Street Address | address1 |
| Street Address 2 | address2 |
| Home Phone # | homephone |
| Work Phone # | workphone |
| Fax Phone # | faxphone |
| Cellular Phone # | mobilephone |
| Email Address | email |
| Job Title | jobtitle |
| Account | accountname |
| Social Security Number | taxnumber |
| Birth Date | dob |

# Configuring Third-Party Key Generation for Oracle Enterprise Data Quality

With the Siebel Data Quality (SDQ) Universal Connector framework the dedup keys used in the candidate selection process are generated by Siebel CRM using a simple token expression. To enable more functionality in key generation you can enable external key generation through a third-party data quality vendor.

To enable third-party key generation for Oracle Enterprise Data Quality, set the Key Generation Process by Third Party parameter to Yes as shown in *Vendor Parameters for Oracle Enterprise Data Quality*.

For implementation, the connector API is `sdq_get_dedup_key`.

# Enabling Support of Multiple Child Entity Deduplication for Oracle Enterprise Data Quality

When using third-party data quality vendors, you can enable Oracle Enterprise Data Quality to perform deduplication on multiple child entities of parent objects, such as Personal Address for the Contact object.

To enable multiple child entity deduplication for Oracle Enterprise Data Quality, set the Support Multiple Child Entities Deduplication parameter to Yes as shown in the following image. For an example of the data format, see *Data Format for Deduplication of Multiple Child Entities*

# Data Format for Deduplication of Multiple Child Entities

To enhance the functionality of Siebel Universal Connector and a third party external key generating vendor, the Siebel Universal Connector API supports hierarchical data format. For example, instead of making multiple calls to the third party vendor for multiple address deduplication, the hierarchical data format sends all child addresses in one payload, thus enabling the connector to make just one call.

The following example shows the data format with the ListOf container used for each child entity.

Note the following:

- The Name field is derived from the parent business component.
- The ListOf container is used for the Address child object.
- The Street field is a join field derived from the child business component.

```
<Data>
 <DriverRecord>
 <Name>…</Name>
 ...
 <ListOfAddress>
 <Address>
 <Street>…</Street>
 ...
 </Address1>
 <Address2>
 <Street>…</Street>
 ...
 </Address2>
 ...
 <AddressN>
 <Street>…</Street>
 ...
 </AddressN>
 </ListOfAddress>
 </DriverRecord>
 <CandidateRecord>
 </CandidateRecord>
</Data>
```

The following example shows the data format for the Contact object.

```
<Data>
```

```
<DriverRecord>
<Contact.Id>12-12ABC1</Contact.Id>
<FirstName>abc</FirstName>
<LastName>abc</LastName>
<ListOfAccount>
<Account>
<Name>A. K. Parker Inc</Name>
<Location>San Mateo</Location>
</Account>
<Account>
<Name>Hibbing Manufacturing</Name>
<Location>San Jose</Location>
</Account>
</ListOfAccount>
<ListOfPersonalAddress>
<PersonalAddress>
<City>York1</City>
<Country>USA</Country>
<Zip>78201</Zip>
<State>AL</State>
<Street> 1200 Davisville Road</Street>
</PersonalAddress>
</ListOfPersonalAddress>
</DriverRecord>
<CandidateRecord>
<Contact.Id>12-BCDD</Contact.Id>
<FirstName>abc1</FirstName>
<LastName>abc1</LastName>
<ListOfAccount>
<Account>
<Name>A. K. Parker Inc</Name>
<Location>San Mateo</Location>
</Account>
</ListOfAccount>
<ListOfPersonalAddress>
<PersonalAddress>
<City>York1</City>
<Country>USA</Country>
<Zip>78201</Zip>
<State>AL</State>
<Street> 1200 Davisville Road</Street>
</PersonalAddress>
</ListOfPersonalAddress>
</CandidateRecord>
<CandidateRecord>
<Contact.Id>12-BCEE</Contact.Id>
<FirstName>abc2</FirstName>
<LastName>abc2</LastName>
<ListOfAccount>
</ListOfAccount>
<ListOfPersonalAddress>
<PersonalAddress>
<City>York3</City>
<Country>USA</Country>
<Zip>78203</Zip>
<State>AL</State>
<Street> 1203 Davisville Road</Street>
</PersonalAddress>
</ListOfPersonalAddress>
</CandidateRecord>
</Data>
```

The following example shows the data format for the Account object.

```
<Data>
 <DriverRecord>
```

```
<Account.Id>12-ABCC</Account.Id>
<Name>Hibbing Manufacturing</Name>
<Location>San Jose</Location>
<ListOfCUTAddress>
<CUTAddress>
<City>York1</City>
<Country>USA</Country>
<Zip>78201</Zip>
<State>AL</State>
<Street> 1200 Davisville Road</Street>
</CUTAddress>
</ListOfCUTAddress>
</DriverRecord>
<CandidateRecord>
<Account.Id>12-AAAA</Account.Id>
<Name>Hibbing Manufacturing1</Name>
<Location>San Jose</Location>
<ListOfCUTAddress>
<CUTAddress>
<City>York1</City>
<Country>USA</Country>
<Zip>78201</Zip>
<State>AL</State>
<Street> 1200 Davisville Road</Street>
</CUTAddress>
<CUTAddress>
<City>York2</City>
<Country>USA</Country>
<Zip>78202</Zip>
<State>AL</State>
<Street> 1201 Davisville Road</Street>
</CUTAddress>
</ListOfCUTAddress>
</CandidateRecord>
<CandidateRecord>
<Account.Id>12-BBBB</Account.Id>
<Name>Hibbing Manufacturing2</Name>
<Location>San Francisco</Location>
</CandidateRecord>
<CandidateRecord>
<Account.Id>12-CCCC</Account.Id>
<Name>Hibbing Manufacturing3</Name>
<Location>San Mateo</Location>
<ListOfCUTAddress>
<CUTAddress>
<City>York1</City>
<Country>USA</Country>
<Zip>78201</Zip>
<State>AL</State>
<Street> 1200 Davisville Road</Street>
</CUTAddress>
</ListOfCUTAddress>
</CandidateRecord>
</Data>
```

# Overview of Configuring Oracle Enterprise Data Quality for Deduplication of Child Entities

Oracle Enterprise Data Quality (EDQ) can be configured to support the deduplication of a parent object's child entities. The steps to do this depend on the complexity of the child entity you want to use. When multiple child entity matching is enabled hierarchical data from Siebel CRM is transformed by the Siebel Connector into a flat record structure so that EDQ can match parent records (such as Contacts, Prospects, or Accounts) using the details of the child records (such as Names, Addresses, Email Addresses, Alternative Phone Numbers, and so on). Siebel Connector can be configured to handle child entities either as scalar (single field) or non-scalar (multi-field) objects.

Scalar groups are used for Siebel business components comprising one field, or for which only one field is mapped in the Data Quality field mappings in the Siebel database. The connector prepares the values for such groups into a simple delimited list of values in a single input attribute for EDQ.

Non-scalar groups are used for more complex child entities comprising multiple fields, such as names and addresses. In this case, the connector prepares multiple records for the same parent to submit to EDQ for matching purposes, each with different child data.

The following is an overview of the steps required to configure Enterprise Data Quality to process additional child entity types. For more information about extending the matching configuration, see *Oracle® Enterprise Data Quality Customer Data Services Pack Customization Guide 11 g Release 1 (11.1.1.7) on* Oracle Technology Network (OTN).

## To extend the matching configuration in Oracle Data Quality (EDQ)

1. In Oracle Data Quality (EDQ), extend the EDQ-CDS project to provide configuration logic to implement deduplication for the child entity type as follows:
    a. Edit the Web service definition (corresponding to the appropriate parent object type) to include the new attributes for the child entity type.
    b. Edit the candidates snapshot to include the new attributes.
    c. Edit the candidates data interface to include the new attributes.
    d. Edit the data interface mapping for the snapshot to include mappings for the new attributes.
    e. Edit the data interface mapping for the Web service to include mappings for the new attributes.
    f. Edit the cluster process to implement any new cluster key generation logic required for the child entity type.
    g. Edit the match process to implement the new matching logic for the child entity type.
2. Modify EDQ Siebel Connector dnd.properties file as follows:
    a. Edit the Multiple child entity groupings section to define the child entity as either a scalar or non-scalar group as appropriate.
    b. Restart the Siebel Server to apply the changes.

# Process of Configuring Additional Child Entities for Deduplication

You can enable or disable support for multiple child entity deduplication with Oracle Enterprise Data Quality (EDQ). By default this feature supports the following child entities: CUT Address, and Personal Address. To enable additional child entities for deduplication, you must first modify the EDQ Siebel Connector and EDQ configuration (including processes, Web services, and matching rules), then perform the steps in this process. To modify the EDQ Connector, see *Overview of Configuring Oracle Enterprise Data Quality for Deduplication of Child Entities*, and *Oracle® Enterprise Data Quality Installation and Upgrade Guide Release 11g R1 (11.1.1.7)* on Oracle Technology Network (OTN), and *Oracle® Enterprise Data Quality Customer Data Services Pack Customization Guide 11g Release 1 (11.1.1.7)* on Oracle Technology Network (OTN).

> **Note:** As an example, this process uses Communication Address (Alternate Email Address) as the child entity for deduplication of the Contact object.

To configure additional child entities for deduplication, perform the following tasks:

1. *Modifying the Data Quality Integration Object for Additional Child Entity Deduplication*
2. *Modifying the Data Quality Business Service for Additional Child Entity Deduplication*
3. *Deploying the Integration Object for Additional Child Entity Deduplication*
4. *Mapping the Third Party Vendor Field for Additional Child Entity Deduplication*
5. *Setting the Third Party Vendor Parameter for Additional Child Entity Deduplication*
6. *Configuring Data Maps for Additional Child Entity Deduplication*
7. *Configuring Oracle Enterprise Data Quality-Siebel Connector for Additional Child Entity Deduplication*

# Modifying the Data Quality Integration Object for Additional Child Entity Deduplication

This task is a step in *Process of Configuring Additional Child Entities for Deduplication*. Use this task to modify the Data Quality integration object for additional child entity deduplication.

## To modify an integration object for additional child entity deduplication

1. In the Web Tools Object Explorer, select Integration Object.
2. Query for the integration object you want to modify. In this example, query for Contact Dedup IO.

> **Note:** The Contact Dedup integration object is used by Oracle Enterprise Data Quality to query for contacts and all child components when performing deduplication. Modify Account Dedup IO if you want to include an additional child entity deduplication for the Account object.

3. In the Integration Component view, add the necessary integration component as shown in the following table.

> **Note:** For this example, you must add the Communication Address integration component.

| Name | External Name | External Name Context | Parent Integration Object Component | XML Container Element | XML Tag |
|---|---|---|---|---|---|
| Contact | Contact | Contact | Not applicable | Not applicable | Contact |
| Personal Address | Personal Address | Contact_INS Personal Address | Contact | ListOfPersonalAddress | PersonalAddress |
| Communication Address | Communication Address | Contact_ Communication Address | Contact | ListOfCommunicationAddre | CommunicationAddress |

4. Select the Communication Address integration component, then in the Integration Component Fields view, add the necessary integration component fields as shown in the following table:

| Name | Data Type | Length | External Name | XML Style | XML Tag |
|---|---|---|---|---|---|
| Id | DTYPE_ID | 30 | Communication AddressId | Element | Contact |
| Address | DTYPE_TEXT | 100 | Alternate Email Address | Element | AlternateEmailAddress |
| Medium Type | DTYPE_TEXT | 30 | Communication Address Medium Type | Element | CommunicationAddressMediumType |
| Name | DTYPE_TEXT | 50 | Communication Address Name | Element | CommunicationAddressName |
| User Type | DTYPE_TEXT | 30 | Communication Address Use Type | Element | CommunicationAddressUse Type |

# Modifying the Data Quality Business Service for Additional Child Entity Deduplication

This task is a step in *Process of Configuring Additional Child Entities for Deduplication*. Use this task to modify the Data Quality business service for additional child entity deduplication.

## To modify the Data Quality business service for additional child entity deduplication

1. In the Web Tools Object Explorer, select Business Service
2. Query for the Data Quality business service: DeDuplication.
3. In the Business Service User Props view, select the necessary user property.

   For example, select Contact Child Dedup Object Name and add Communication Address to the list as follows:

   - Name: Contact Child Dedup Object Name
   - Value: Personal Address;Communication Address.

   **Note:** List all child Dedup objects for Contact and separate each object entry with a semicolon (;).

# Deploying the Integration Object for Additional Child Entity Deduplication

This task is a step in *Process of Configuring Additional Child Entities for Deduplication*. Use this task to deploy the modified repository objects for additional child entity deduplication.

## To deploy the integration object for additional child entity deduplication

1. Stop the Siebel Server and deliver the modified integration object and business service to your Siebel runtime repository, then deploy the integration object by doing the following:
   a. Select the modified integration object in Web Tools, right-click and choose Undeploy, then step off the record to retain the changes.
   b. Right-click again on the integration object and choose Deploy to Runtime Database, then step off the record.
2. Start the Siebel Server.

# Mapping the Third Party Vendor Field for Additional Child Entity Deduplication

This task is a step in *Process of Configuring Additional Child Entities for Deduplication*. Use this task to map the third party vendor field for additional child entity deduplication.

### To map the third party vendor field for additional child entity deduplication

1. In Oracle Customer Hub (UCM), navigate to Administration - Data Quality, then Third Party Administration.
2. From the Third Party Administration screen, select EDQ in the Vendor view.
3. Click the BC Vendor Field Mapping tab, then in the BC Vendor Field Mapping view, add a new entry for the Business Component, and Operation fields. For this example enter the following:
    - Business Component: Communication Address
    - Operation: Deduplication
4. In the Field Mappings view, enter the correct business component field. For this example, enter the following:
    - Business Component Field: Address
    - Mapped Field: email

    For a list of all vendor field mappings, see *Oracle Enterprise Data Quality Field Mappings for Siebel Business Components*.

## Setting the Third Party Vendor Parameter for Additional Child Entity Deduplication

This task is a step in *Process of Configuring Additional Child Entities for Deduplication*. Use this task to set the third party vendor parameter for additional child entity deduplication.

### To set the third party vendor parameter for additional child entity deduplication

1. In Oracle Customer Hub (UCM), navigate to Administration - Data Quality, then Third Party Administration.
2. From the Third Party Administration screen, select EDQ in the Vendor view.
3. Click on the Vendor Parameter tab, and add a new record to specify the DeDup Record Type for the new child component. For this example, enter the following:
    - Vendor Parameter Name: Communication Address DeDup Record Type
    - Value: emailmatch

    For a list of all vendor parameters, see *Vendor Parameters for Oracle Enterprise Data Quality*.

## Configuring Data Maps for Additional Child Entity Deduplication

This task is a step in *Process of Configuring Additional Child Entities for Deduplication*.

To use a new child entity for matching in Oracle Customer Hub (UCM) processes, you must add the new integration component and integration component fields to Oracle Customer Hub (UCM) integration objects (such as CIFContact, SwiPersonIO, or UCMContactSourceDataAndHistory). Additionally, to allow Oracle Customer Hub (UCM) to send the new fields to Oracle Enterprise Data Quality for matching, you must map the new fields from the Oracle Customer Hub (UCM) integration object to the Oracle Enterprise Data Quality integration object.

To perform the following steps you must have performed all preceding steps in this process to modify all required objects to include the new child integration component and integration component fields, and also the modified integration objects must have been redeployed to the Siebel run-time database.

Use this task to configure data maps for additional child entity deduplication.

## To configure data maps for additional child entity deduplication

**Note:** EAI Data Maps can be Workspace enabled in your development environment. If they have been Workspace enabled in your development environment and you are working in that environment, then you can only modify them in an editable Workspace. You do not need an editable Workspace to create and edit EAI Data Maps in your Production environment.

1. In Oracle Customer Hub (UCM), and navigate to Administration - Integration, then Data Map Editor.
2. In the Integration Object Map view, query for the required integration object map.

   For this example, use the following query: UCM Contact*DQ IO.
3. Modify the required maps.

   For this example, modify the following maps for Contact Deduplication:
   - UCM Contact CIF IO to DQ IO (used by the UCM Real-time Process).
   - UCM Contact SDH IO to DQ IO (used by the UCM Batch Process).
   - UCM Contact SOAP IO to DQ IO (used by the UCM Soap Process).
   - UCM Contact Match IO to DQ IO (used by the Person Match Web Service).
4. For each data map in this example, you must use the Integration Component Map view to add the Contact_Communication Address integration component. See the following table:

| Name | Source Component Name | Target Component Name | Source Search Specification |
|---|---|---|---|
| Contact | Contact | Contact | Not applicable |
| Contact_INS Personal Address | Contact INS Personal Address | Personal Address | Not applicable |
| Contact Account | Contact Account | Contact | IsPrimaryMVG = 'Y' |
| Contact_Communication Address | Contact Communication Address | Communication Address | Not applicable |

5. Select the newly created integration component map, Contact_Communication Address, then in the Integration Field Map view add the mapping for an alternate email address using the following example:
   - Source Expression: [Alternate Email Address]
   - Target Field Name: Address

6. Repeat Step 4 and Step 5 for each additional data map.

# Configuring Oracle Enterprise Data Quality-Siebel Connector for Additional Child Entity Deduplication

This task is a step in *Process of Configuring Additional Child Entities for Deduplication*. Use this task to configure Oracle Enterprise Data Quality-Siebel Connector for additional child entity deduplication.

For more information on the Scalar groups area of the dnd.properties file, see *Overview of Configuring Oracle Enterprise Data Quality for Deduplication of Child Entities*.

## To configure the Oracle Enterprise Data Quality-Siebel Connector for additional child entity deduplication

1. Locate the dnd.properties file for Oracle Enterprise Data Quality-Siebel Connector.
2. In the properties file, locate the Multiple child entity groupings section, and under the Scalar groups to be stored as delimited lists in a single field sub-section, add the following scalar group:
   - group.email. Add a property value of **CommunicationAddress**.
   - group.email.concat. Add a property value of **email**.
   - group.email.delimiter. Add a property value of **|**.

   With these settings, Oracle Enterprise Data Quality-Siebel Connector appends all alternate email addresses in the CommunicationAddress child entity and sends them as a single field (in an email attribute) to the Oracle Enterprise Data Quality (EDQ) server for matching.

   > **Note:** Support for matching against multiple child entities was added in EDQ-CDS version 9.0.2 but it is recommended that the latest compatible version of the software is used. See the Enterprise Data Quality Certification Matrix for details about compatible software versions: *http://docs.oracle.com/cd/E48549_01/doc.11117/e40736/cdsis.htm#sthref4*.

# Sample Message from Siebel CRM to EDQ Siebel Connector for Additional Child Entity Deduplication

The following is a sample message with multiple email addresses sent from Siebel CRM. Note that there are is email address at the parent contact level (DriverRecord). The list of alternate email addresses are sent in the child entity (Communication Address) which maps to the Email attribute in EDQ.

```
<Data>
 <DriverRecord>
 <firstname>Mike</firstname>
 <lastname>Jones</lasttname>
 <workphone>1234567890</workphone>
 <ListofPersonalAddress>
 <PersonalAddress>
 <address1>100 First Street</address1>
 <city>San Jose</city>
```

```
<state>CA</state>
<country>USA</country>
</PersonalAddress>
<ListofCommunicationAddress>
<CommunicationAddress>
<email>mike@yahoo.com</email>
<usage>Personal</usage>
</CommunicationAddress>
<CommunicationAddress>
<email>mike@stanford.edu</email>
<usage>Campus</usage>
</CommunicationAddress>
<CommunicationAddress>
<email>mike@oracle.com</email>
<usage>Business</usage>
</CommunicationAddress>
</ListofCommunicationAddress>
</DriverRecord>
</Data>
```

# Overview of Configuring Social Media Child Objects for Matching

With Siebel Innovation Pack 2014 you can enable the Contact object child entity "SM Contact Social Profile" for matching in Oracle Customer Hub (UCM). Configuration tasks include:

## Prerequisite

To perform this process, you must first have completed the following task:

- Configure the EDQ Connector, as described in: *Overview of Configuring Oracle Enterprise Data Quality for Deduplication of Child Entities*

Then perform the tasks listed in the process.

To configure additional child entities for deduplication, perform the following tasks:

1. *Mapping the Third Party Vendor Field for Additional Child Entity Deduplication*
2. *Setting the Third Party Vendor Parameter for Additional Child Entity Deduplication*
3. *Configuring Data Maps for Additional Child Entity Deduplication*
4. *Configuring Oracle Enterprise Data Quality-Siebel Connector for Additional Child Entity Deduplication*

# Sample Message from Siebel CRM to EDQ Siebel Connector for Social Media Child Object Matching

The following is a sample message with multiple Contact Social Profiles sent from Siebel CRM to EDQ Siebel Connector for Contact matching.

```
<Data>
```

```
    <DriverRecord>
    <Contact.Id>12-12ABC1</ContactId>
    <firstname>Mike</firsttname>
    <lastname>Jones</lasttname>
    <ListofSMContactSocialProfile>
    <SMContactSocialProfile>
    <Author>Mike Jones</Author>
    <Community>Facebook</Community>
    </SMContactSocialProfile>
    <SMContactSocialProfile>
    <Author>Mike Jones</Author>
    <Community>Google</Community>
    </SMContactSocialProfile>
    <SMContactSocialProfile>
    <Author>jones</Author>
    <Community>Instagram</Community>
    </SMContactSocialProfile>
    <SMContactSocialProfile>
    <Author>@jones</Author>
    <Community>Twitter</Community>
    </SMContactSocialProfile>
    <SMContactSocialProfile>
    <Author>Mike Jones</Author>
    <Community>YouTube</Community>
    </SMContactSocialProfile>
    </ListofSMContactSocialProfile>
    </DriverRecord>
  </Data>
```

# 7 Configuring Data Quality with Oracle Data Quality Servers

## Configuring Data Quality with Oracle Data Quality Servers

This chapter describes the data quality configuration that you can perform for Oracle Data Quality Matching Server and Oracle Data Quality Validation Server. It covers the following topics:

- *Process of Installing the Oracle Data Quality Matching Server*
- *Upgrading to Informatica Identity Resolution 9.01*
- *Configuring Oracle Data Quality Matching Server*
- *Universal Connector Parameter and Field Mapping Values for Oracle Data Quality Matching Server*
- *Configuring a New Field for Real-Time Data Matching*
- *Loading Data Incrementally*
- *Process of Configuring Data Synchronization Between Siebel CRM and Oracle Data Quality Matching Server*
- *Siebel CRM DQ Sync Action Sets*
- *Configuring Deduplication Against Multiple Addresses*
- *Configuring Multiple Language Support for Data Matching*
- *Configuring Multiple Mode Support for Data Matching*
- *Sample Configuration Files for Oracle Data Quality Matching Server*
- *Process of Installing the Oracle Data Quality Address Validation Server*
- *Configuring Siebel CRM for the Oracle Data Quality Address Validation Server*
- *Universal Connector Parameter and Field Mapping Values for Oracle Data Quality Address Validation Server*
- *Sample Configuration File for Oracle Data Quality Address Validation Server*
- *Siebel CRM ISSSYNC Action Sets*
- *Activating Action Sets in Your Siebel Application*
- *Sample SQL Scripts*
- *Sample SiebelDQ.sdf File*

**Note:** You must be familiar with Siebel Tools before performing some of the data quality configuration tasks. For more information about Siebel Tools, see *Using Siebel Tools* and *Configuring Siebel Business Applications* .

# Process of Installing the Oracle Data Quality Matching Server

The Oracle Data Quality Matching Server provides real-time and batch data matching functionality using licensed third-party Informatica Identity Resolution 953 (IIR) software. The Oracle Data Quality Matching Server connector uses the Universal Connector in a mode where match candidate acquisition takes place within the Oracle Data Quality Matching Server. Since the match keys are generated and stored within the Oracle Data Quality Matching Server, key generation and key refresh operations are eliminated within Siebel CRM.

To install the Oracle Data Quality Matching Server for data matching, perform the following tasks:

1. *Setting Up the Environment and the Database*
2. *Installing Oracle Data Quality Matching Server*
3. *Creating Database Users and Tables for Oracle Data Quality Matching Server*
4. *Configuring Oracle Data Quality Matching Server*
5. *Modifying Configuration Parameters for Oracle Data Quality Matching Server*
6. *Deploying Workflows for Oracle Data Quality Matching Server Integration*
7. *Initial Loading of Siebel Data into Oracle Data Quality Matching Server Tables*

## Setting Up the Environment and the Database

This topic describes the prerequisites that are needed before starting to install Oracle Data Quality Matching Server for data matching, and also how to set up an Oracle database for Oracle Data Quality Matching Server. This task is a step in *Process of Installing the Oracle Data Quality Matching Server*.

### Java Runtime Environment

The installation and operation of the Oracle Data Quality Matching Server is controlled by a Java application called the Console Client. The Console Client can be run on any operating system that supports Java 1.4 or later, and Java Help 1.1 is required. In order to run the Workbench, the Java Runtime Environment (JRE) is required.

JRE must be installed on the same computer as the Console Client. Before running the Console Client, ensure that the PATH and CLASSPATH environment variables have been set up for the correct Java and Javahelp installations.

For example, on a Win32 client:

```
SET CLASSPATH=%JAVAHELP_HOME%\jhall.jar
SET PATH=%PATH%;%JAVA_HOME%\bin
```

On UNIX:

```
SSAJDK="/usr/java/jdk1.5.0_14"
CLASSPATH="/export/home/qa1/jh2_0/javahelp/lib/jhall.jar"
```

On UNIX, you set the PATH and CLASSPATH environment variables in the ssaset script file.

### Network Protocol

Clients and Servers require a TCP/IP network connection. This includes DNS, which must be installed, configured and available (and easily contactable). The following paths (or their equivalents) must be correctly set up: `/etc/hosts`, `/etc/resolv.conf` and `/etc/nsswitch.conf`. Reverse name lookups must yield correct and consistent results.

## ODBC Driver

The Oracle Data Quality Matching Server uses Open Database Connectivity (ODBC) to access source and target databases. ODBC Drivers for specific databases must be installed and working. Installing and configuring ODBC drivers is operating system and database dependent. Unless a driver is provided by Oracle Data Quality Matching Server (as is the case for an Oracle database), you must follow the instructions provided by your database manufacturer in order to install them. On Windows operating system, navigate to Control Panel, Administrative Tools, and then Data Sources (ODBC) to create a DSN and associate it with a driver and database server.

At run time, the database layer attempts to load an appropriate ODBC driver for the type of database to be accessed. The name of the driver is determined by reading the odbc.ini file and locating a configuration block matching the database service specified in the connection string. For example, the database connection string `odb:99:scott/tiger@ora920` refers to a service named ora920. A configuration block for [ora920] looks similar to the following:

```
[ora920]
ssadriver = ssaoci9
ssaunixdriver = ssaoci9
server = ora920.myexample.com
```

A configuration block has the following syntax:

```
[Service_Name]
DataSourceName = ODBC_DSN
ssadriver = ODBC_Driver
ssaunixdriver = ODBC_UNIX_Driver
server = Native_DB_Service_Name
```

*Creating Database Users and Tables for Oracle Data Quality Matching Server* shows the databases supported by Oracle Data Quality Matching Server, describes the ODBC drivers required for different operating systems, and shows example odbc.ini configurations.

> **Note:** Oracle Data Quality Matching Server provides a custom driver for the Oracle database that is installed during the installation of the product. Oracle Data Quality Matching Server does not use the standard driver shipped with the Oracle DBMS.

# Installing Oracle Data Quality Matching Server

This task is a step in *Process of Installing the Oracle Data Quality Matching Server*. The following tasks describe the steps involved in installing Oracle Data Quality Matching Server on Microsoft Windows and on UNIX respectively.

- *Installing Oracle Data Quality Matching Server on Microsoft Windows*
- *Installing Oracle Data Quality Matching Server on UNIX*

> **Note:** License key information for the Oracle Data Quality Matching Server is included in the product media pack on Oracle Software Delivery Cloud.

## Installing Oracle Data Quality Matching Server on Microsoft Windows

Use the following procedure to install Oracle Data Quality Matching Server on Microsoft Windows.

**Note:** Installation is the same no matter what version of Informatica Identity Resolution you are installing.

## To install Oracle Data Quality Matching Server on Microsoft Windows

1. Run setup from the root directory of the installed product media pack (which you downloaded from Oracle Software Delivery Cloud), and follow the on-screen prompts to install the following options:
   - Install License Server
   - Install Informatica Product

   **Note:** You must install these options in the order that they are displayed.
2. Select Install License Server, click Next to continue, then do the following:
   a. Browse to the installation directory where you want to install the License Server, then click Next.
   b. Enter the host name and port number for the License Server.
   c. Verify the installation summary details on the next screen that displays, then click Install.
   d. When installation is complete, you are prompted to start the License Server. Click No to close the prompt, then Finish to return to the main installer window.
   e. Copy the OEM license key file downloaded from Oracle Software Delivery Cloud to the following location:

      ```
      <Drive:>\InformaticaIR\licenses
      ```

   f. Set the environment variable SSALI_MZXPQRS to STANISLAUS (system variable).
   g. Start the License Server by selecting, for example, the following:

      Programs, Informatica, Identity Resolution V2.8.07 (InformaticaIR), Informatica License Server, and then Start.

3. Select the Install Informatica Product from the main installer window, click Next to continue, then do the following:

    a. When prompted to specify the path to the OEM license, browse to the `[installation_media_directory]\data\file1003.dat` file, and then click Next to continue.
    b. Enter the host name and port number for the License Server (or accept the default), then click Next.
    c. Browse to the installation directory where you want to install Informatica Identity Resolution, then click Next.
    d. The next screen displays a list of components, click Select All, and then Next.
    e. The next screen displays an installation summary of products and modules that you want to install. Review the details and click Next to confirm that they match your requirements.

       Make sure to add XML Synchronization server at port 1671. This server is not set by default. Click Next when done.
    f. Select default port values for all servers.
    g. On the next screen, enter database information, as follows:

        - **Service Name:** Enter the database service name on Informatica Identity Resolution. This is used when configuring SIEBEL instances.
        - **ODBC Data Source Name:** Enter the ODBC Connect String name if using ODBC (the ODBC Data Source name is required only when connecting through ODBC).
        - **ODBC Driver:** Select the applicable database driver from the drop-down list (the ODBC driver name must be provided even when ODBC is not being used).
        - **Native Service:** Enter the name for the database connection as defined in `dB Client\ Server` utilities (for example: for Oracle an databases, this is the TNS entry name).

          Example settings when using an Oracle database are:

```
 Service Name: target
ODBC Data Source Name:
 ODBC Driver: Oracle 9 (or later) client software
 Native Service Name: <tns_entry>
```

    > **Note:** All configuration information entered in this step is written to the odbc.ini file. *Creating Database Users and Tables for Oracle Data Quality Matching Server* shows some example odbc.ini configurations.

    h. Click Next to start the installation.
    i. Click Finish to complete.
4. Post installation, do the following:

    a. Install the hot fix on top of the Base Installer for Informatica Identity Resolution 2.8.07. Make sure that you apply the latest Informatica Identity Resolution fix, which is available on Oracle Software Delivery Cloud.

```
C:\InformaticaIR\bin>version
SSA-NAME3 v2.8.07 (FixL106)
SSA-NAME3 Extensions v2.8.07 (FixL106)
Data Clustering Engine v2.8.07 (FixL106)
Informatica Identity Resolution v2.8.07 (FixL106 + FixL113 + FixL114 + FixL120
+ FixL123 + FixL124 + FixL125 + FixL126 + FixL127 + FixL134 + FixL136 + FixL140
+ FixL141 + FixL145 + FixL147 + FixL148)
```

b. Rename xsserv.xml.org located in `<drive>\InformaticaIR\bin` to xsserv.xml. This file has a sample format. Change it to match the following:

```
<server xmlns="_http://www.identitysystems.com/xmlschema/iss-version-1/
xmlserv">
 <mode>generic</mode>
 <rulebase>odb:0:db_username/db_password@ISS_connectstring</rulebase>
</server>
```

> **Note:** If you do not make these changes to xsserv.xml, then errors might occur using legacy SIEBEL-ISS Sync workflows.

## Installing Oracle Data Quality Matching Server on UNIX

Use the following procedure to install Oracle Data Quality Matching Server on UNIX.

> **Note:** Installation is the same no matter what version of Informatica Identity Resolution you are installing.

### To install Oracle Data Quality Matching Server on UNIX

1. The following prerequisites must be met:
   a. Obtain an installable ISO from Informatica or Oracle Software Delivery Cloud.
   b. Make sure to mount the ISO to a suitable location which is accessible to the user who is installing Informatica products.
   c. Ensure that the required ODBC entries (if applicable) are created.
   d. Ensure to add TNS entries pointing to the Informatica Identity Resolution Database (target database) for system configurations.
   e. The Informatica Installer requires a UNIX GUI for installation and product administration. If necessary, contact your system administrator to make sure that the corresponding applications and X-Server have been enabled for easy access.
2. After ensuring the installer location and GUI access, start the installer from the mounted location using the following command:

```
./install
```

   The Informatica Installer window opens with three options. You must install the three options in the order that they are displayed.

3. Select the Install License Server from the installer, click Next to continue, then do the following:

   a. Select the path where you want to install the license server, then click Next.
   b. Enter the port number for the License Server on the next screen that displays. You can accept the default (if available), or choose to change the port. Click Next when done.
   c. Verify the installation summary details on the next screen that displays, then click Install.
   d. When installation is complete, you are prompted to start the License Server. Click No, and then Finish to return to the main installer window. You must start the License Server only when the license file is available.
   e. Copy the OEM license key file downloaded from Oracle Software Delivery Cloud to the following location:

      `<Drive:>/InformaticaIR/licenses`

   f. Export the environment variable SSALI_MZXPQRS to STANISLAUS (system variable) before proceeding to the next step.
   g. Start the License Server:

      - Start an xterm / ssh session.
      - Change to bash (Bourne Shell)
      - Copy the license file to `<installation_folder>/licenses`

   h. Set common environment variables by sourcing idsset script located at `<IIR_Installation_Folder>/env`. For example:

      ```
      . ./idsset
      ```

   i. Set the environment variables required to start the License Server by sourcing script lienvs located at `<IIR_Installation_Folder>/env`. For example:

      ```
      . ./lienvs
      ```

   j. Start the License Server using the following command: `$SSABIN/liup`.

4. Select Install Informatica Product from the main installer window, click Next to continue, then do the following:

    a. When prompted to specify the path to the OEM license, browse to the `[installation_media_directory]/data/file1003.dat` file, and then click Next to continue.
    b. Enter the License Server port number or accept the default, then click Next.
    c. The next screen displays a list of components. Licensed components have an editable check-box. Select the check box beside the required components and populations, and then click Next.
    d. The next screen displays a summary of selected options. Verify the details, then click Next.
    e. On the next screen, select or set servers and their ports, then click Next. If a port is already in use, you must change it.
    f. On the next screen, enter database information:

        - **Service Name:** Enter the database service name on Informatica Identity Resolution (this is used when configuring SIEBEL instances).
        - **ODBC Data Source Name:** Enter the ODBC Connect String name if using ODBC (the ODBC Data Source name is required only when connecting through ODBC).
        - **ODBC Driver:** Select the applicable database driver from the drop-down list (the ODBC driver name must be provided even when ODBC is not being used).
        - **Native Service:** Enter the name for the database connection as defined in `dB Client/Server` utilities (for example: for Oracle an databases, this is the TNS entry name).

        Example settings when using an Oracle database are:

        ```
         Service Name: target
        ODBC Data Source Name:
         ODBC Driver: Oracle 9 (or later) client software
         Native Service Name: <tns_entry>
        ```

        Click Next to continue.

        > **Note:** All configuration information entered in this step is written to the odbc.ini file. *Creating Database Users and Tables for Oracle Data Quality Matching Server* shows some example odbc.ini configurations.

    g. The next screen displays an installation summary of products and modules that you want to install. Verify the details and confirm that they match your requirements.
    h. Click Install to start the installation.
    i. Click Finish to complete.

5. Post installation, rename xsserv.xml.ori located in `<IIR_Installation_Folder>/bin` to xsserv.xml. Change the contents of this sample file as follows:

    a. Change <mode> to generic
    b. Add rulebase details.

    For example:

    ```
    <server xmlns="_http://www.identitysystems.com/xmlschema/iss-version-1/xmlserv">
    <mode>generic</mode>
    <rulebase>odb:0:db_username/db_password@ISS_connectstring</rulebase> </server>
    ```

    > **Note:** If you do not make these changes to xsserv.xml, then errors might occur using legacy SIEBEL-ISS Sync workflows.

# Creating Database Users and Tables for Oracle Data Quality Matching Server

Creating database users and tables involves executing a number of scripts on the Oracle Data Quality Matching Server database. The scripts that you must execute are located in the Oracle Data Quality Applications media pack on Oracle Software Delivery Cloud. Once the media pack is downloaded and installed, the scripts are located in the `InformaticaIR/dbscript/ora` folder. For example (idsuseru.sql):

```
C:/InformaticaIR/dbscript/ora/idsuseru.sql
```

You must open these scripts and modify them as required, depending on the database that you are using. For example, complete the steps in the following procedure to create database users and database tables for Oracle Data Quality Matching Server if using an Oracle database. Note the following:

- The procedure is similar if using Microsoft SQL Server, UDB, or DB2 on OS/390. However, you must modify the SQL scripts according to the database that you are using.
- The procedure is also similar whether creating database users and database tables for Oracle Data Quality Matching Server on Microsoft Windows or on UNIX.
- When setting up the database for Oracle Data Quality Matching Server on UNIX, you must set TNSNAmes.ora with an entry to the target database (Oracle Data Quality Matching Server database), and perform connectivity testing using SQLPLUS if required.

This task is a step in *Process of Installing the Oracle Data Quality Matching Server*.

## To create database users and tables for Oracle Data Quality Matching Server if using an Oracle database

1. Log in to the database as database administrator, then execute the idsuseru.sql script to create a new database user with appropriate privileges to create and update Oracle Data Quality Matching Server tables.

   **Note:** You must be logged in as database administrator to execute idsuseru.sql.

2. Log in to the database as the new database user (created in Step 1 with appropriate privileges to create and update Oracle Data Quality Matching Server tables), then execute the following SQL scripts to create other

Oracle Data Quality Matching Server database tables, such as IDT and IDX tables. You can execute the following SQL scripts in any order:

> **Note:** IDT tables store the copy of source records in the Oracle Data Quality Matching Server database. IDX tables store the index keys for IDT tables. Each IDT table can have one or more IDX tables associated with it.

a. Execute idstbora.sql to create control tables for the Oracle Data Quality Matching Server.
b. Execute updsyncu.sql to create database objects required by the Oracle Data Quality Matching Server to synchronize data in ID tables with updates to user source tables.

   Run this script on all databases containing user source tables that require synchronization, and also before loading any ID tables that require synchronization.
c. Execute updsynci.sql to create database objects required by the Oracle Data Quality Matching Server to synchronize data in ID tables with updates to user source tables.

   Run this script on the database which will contain IDTs, and also before loading any ID tables that require synchronization.
d. Execute updsyncg.sql to create database objects required by the Oracle Data Quality Matching Server to synchronize data in SSA-ID tables with updates to user source tables.

   This script will create public synonyms for the Oracle Data Quality Matching Server objects created on user source table databases. This script must be run by someone (for example, the database administrator) who has the privilege to CREATE PUBLIC SYNONYM. Run this script after running updsyncu.sql. Use the same userid to run updsynci.sql as you did to run updsyncu.sql.

   > **Note:** You must be logged in as database administrator to execute updsyncg.sql.

The following table describes some example odbc.ini configurations.

| Database | Description | Example odbc.ini Configurations |
|---|---|---|
| Oracle Database 10*g* | The Oracle database driver works out-of-the box and is named `%SSABIN%\ssaoci{8\|9}.dll` on Windows, and `$SSABIN/libssaoci{8\|9}.s{o\|l}` on UNIX. There are no special setup requirements, other than adding configuration blocks to your odbc.ini file. The ODBC_Driver name can be either ssaoci8 or ssaoci9. The former must be used with Oracle 8 client libraries and does not support Unicode data. The latter can be used with Oracle 9 (or later) client libraries and supports Unicode access.<br><br>When using the ssaoci9 driver with Oracle Database 10*g* client software, the connectivity test might fail on some UNIX operating systems. This occurs because the driver has been linked with libclntsh.so.9.0, which is not distributed with Oracle Database 10*g*. Oracle normally provides backward compatibility by adding symbolic links to redirect requests for older versions of the library to the current version. Unfortunately, by default, this practice is restricted to minor versions only (for example, 9.0-9.2). To overcome the problem, locate the appropriate Oracle lib directory (lib, lib32, or lib64) and add a symbolic link. For example:<br><br>`cd $ORACLE_HOME/lib32`<br>`ln -s ./libclntsh.so libclntsh.so.9.0` | `[ora10g]`<br>`ssadriver = ssaoci9`<br>`ssaunixdriver = ssaoci9`<br>`server = ora10g.mynet8tns.name` |

| Database | Description | Example odbc.ini Configurations |
|---|---|---|
| Universal Database (UDB) | UDB must be installed prior to the installation of Oracle Data Quality Matching Server.<br><br>IBM provides ODBC drivers for both Windows and UNIX operating systems, named db2cli and db2 respectively.<br><br>For more information about the db2cli and db2 drivers, see the appropriate UDB manuals for full details. | `[test-udb]`<br>`DataSourceName = udb8`<br>`ssadriver = db2cli`<br>`ssaunixdriver = db2`<br>`server = UDB_database_alias` |
| Microsoft SQL Server | Microsoft provides a Windows ODBC driver named sqlsrv32. It is configured by adding a new Data Source Name (DSN) using Control Panel, Administrative Tools, Data Sources (ODBC).<br><br>For more information about the sqlsrv32 driver, see the appropriate Microsoft manuals for specific details.<br><br>The ODBC_Driver name is sqlsrv32 and the Native_DB_Service is the server name (-S parameter of the osql and bcp utilities).<br><br>The SQL Server Native Client (sqlncli.dll) can be used as an alternative driver. | `[production]`<br>`DataSourceName = msq2003`<br>`ssadriver = sqlsrv32`<br>`server = mySQLServer` |
| Sybase | For more information about the sybdrvodb drivers, see the appropriate Sybase manuals for installation specifics. | `[production]`<br>`DataSourceName = ase150`<br>`ssadriver = sybdrvodb`<br>`ssaunixdriver = sybdrvodb`<br>`server = mySybaseServer` |

## Testing Connectivity

Use the `dblist` utility to test your ODBC configuration by connecting to a database whose connection string is provided with the `-d` parameter. An example of the output associated with a successful connection follows:

```
$SSABIN/dblist -c -dodb:99:ssa09/SSA09@ora920
Maximum connections per module: 1024
Linked databases: odb: sdb:
Driver Manager: 'Identity Systems ODBC Driver Manager 1.2.2.3'
ODBC Driver: 'ssaoci9 SSADB8 2.7.0.00MSVC60 Jun 8 2006 17:26:56'
DBMS Name: 'Oracle DBMS (9.2.0.6.0)'
Native DB type: 'ora'
```

# Configuring Oracle Data Quality Matching Server

The following tasks describe the steps involved in configuring Oracle Data Quality Matching Server for data matching on Microsoft Windows and on UNIX respectively.

- *Configuring Oracle Data Quality Matching Server on Microsoft Windows*
- *Configuring Oracle Data Quality Matching Server on UNIX*

This task is a step in *Process of Installing the Oracle Data Quality Matching Server*.

## Configuring Oracle Data Quality Matching Server on Microsoft Windows

Use the following procedure to configure Oracle Data Quality Matching Server on Microsoft Windows.

### To configure Oracle Data Quality Matching Server for data matching on Microsoft Windows

1. If required, modify the odbc.ini file located at `<drive>:\<IIR_Installation_Folder>\InformaticaIR\bin\` to contain the ODBC connection string of your target database, for example, as follows:

   ```
   [Target]
   ssadriver=ssaoci9
   server=qa19b_sdchs20n519
   ```

   *Creating Database Users and Tables for Oracle Data Quality Matching Server* describes the ODBC drivers required for different operating systems.

   > **Note:** For an Oracle database, the server parameter specifies a connect string from the tnsnames.ora file (which is the network configuration file of the Oracle database client). For other databases, the server contains the ODBC datasource name (DSN).

   The database information that you enter when installing Oracle Data Quality Matching Server is reflected in the odbc.ini file. If all values are correct and you do not want to make any changes to the database information, then you can skip this step.
2. Copy the SiebelDQ.sdf file to the following (IIR server) folder location:

   ```
   <Drive>:\<IIR_Installation_Folder>\InformaticaIR\ids
   ```

   > **Note:** For an example SDF file, see *Sample SiebelDQ.sdf File*.

3. To use the XML Sync Server instead of the External Business Components for Informatica Identity Resolution, then activate or deactivate the following ports located in `<Drive>:\<IIR_Installation_Folder>\env\isss.bat`.

   ```
   ::set SSA_XSPORT=1671
   ::set SSA_XSHOST=localhost:1671
   ```

   Removing the double colon from the beginning of the line activates the process listening on the ports:

   ```
   set SSA_XSHOST=localhost:1671
   set SSA_XSPORT=1671
   ```

   > **Note:** For Informatica Identity Resolution Version 2.7, you turn on the XML Sync Server by modifying the idsenvs.bat file located in `<Drive>:\<ISS Installation Folder>\iss2704s\bin`.

4. Create a tmp folder for the IIR Synchronizer Workflow Log in `<Drive>:\<IIR_Installation_Folder>\InformaticaIR\`. For example:

   ```
   C:\InformaticIR\tmp
   ```

> **Note:** If you install Oracle Data Quality Matching Server on a different drive (other than `c:\`), you must modify the ISSErrorHandler workflow in your Siebel application to specify the correct log folder. Other modifications that must be made if you install Oracle Data Quality Matching Server on a drive other than `c:\` include modifying action sets and the location where you deploy the XML files.

5. Start the IIR Server by navigating to, for example, the following:
   Programs, Informatica, Identity Resolution V2.8.07 (InformaticaIR), Informatica Identity Resolution, Informatica IR Server - Start(Configure Mode)

   > **Note:** You can also start the Informatica Identity Resolution server from the command prompt using the idsup command.

6. Start the IIR Console Client (in Admin Mode) by navigating to, for example, the following:
   Programs, Informatica, Identity Resolution V2.8.07 (InformaticaIR), Informatica Identity Resolution, Informatica IR Console Client - Start(Admin Mode)
7. Create a new system in IIR using SiebelDQ.sdf.
   The system that you create in IIR (Console Client, Admin Mode) will hold all the IDT and IDX database tables.
8. When the system is created (initially, it will be empty), run LoadIDT from the IIR Console Client. For more information, see *Initial Loading of Siebel Data into Oracle Data Quality Matching Server Tables*.

## Configuring Oracle Data Quality Matching Server on UNIX

Use the following procedure to configure Oracle Data Quality Matching Server on UNIX.

### To configure Oracle Data Quality Matching Server for data matching on UNIX

1. Copy the most recent version of the shared library libssaiok.so (libssaiok.sl on HP-UX) to the SSA-NAME3 bin directory.
   If the version packaged with IIR is more recent than the one packaged with SSA-NAME3, copy the ssaiok shared library from the IIR server distribution to the SSA-NAME3 bin directory as follows:

   ```
   cp $SSATOP/common/bin/libssaiok.* $SSAN3V2TOP/bin
   ```

   No action is required if the version packaged with IIR is older than the one packaged with SSA-NAME3.
2. Set the shared library path according to your operating system.
   The following table shows examples of shared library paths.

| Operating System | Shared Library Path |
|---|---|
| Linux and Oracle Solaris | `LD_LIBRARY_PATH="$SSABIN:$SSANM3BIN:<DBMS shared object location>" export LD_LIBRARY_PATH` |
| HP-UX | `SHLIB_PATH="$SSABIN:$SSANM3BIN:<DBMS shared object location>" export SHLIB_PATH` |
| AIX | `LIBPATH="$SSABIN:$SSANM3BIN:<DBMS shared object location>"` |

| Operating System | Shared Library Path |
|---|---|
| | `export LIBPATH` |

3. If required, modify the odbc.ini file to contain the ODBC connection string of your target database:

   a. Copy the odbc.ini.ori file located in the `$SSATOP/bin` folder, and rename it odbc.ini.
   b. Edit the odbc.ini to contain the ODBC connection string of your target database, for example, as follows:

```
[Target]
ssaunixdriver=ssaoci9
server=<TNS_entry_name_from_tnsnames.ora>
```

   For an Oracle database, the server parameter specifies a connect string from the tnsnames.ora file (which is the network configuration file of the Oracle database client). For other databases, the server contains the ODBC datasource name (DSN). Most UNIX installations do not need the ODBC DSN, but if required, parameters change accordingly:

```
 [Target]
DataSourceName=ODBC_DNS_Name_Pointing_to_ISS_DB
ssaunixdriver=<ssaoci9>
```

   *Creating Database Users and Tables for Oracle Data Quality Matching Server* describes the ODBC drivers required for different operating systems.

   The database information that you enter when installing Oracle Data Quality Matching Server is reflected in the odbc.ini file. If all values are correct and you do not want to make any changes to the database information, then you can skip this step.

4. Copy the System Definition File (SDF) to the UNIX server.

   Make sure that the SDF file is compressed before using FTP to copy it to the UNIX server. You must use the -a switch to extract a file on a UNIX server, for example, as follows:

```
unzip - sysdeffile.zip
```

# Modifying Configuration Parameters for Oracle Data Quality Matching Server

The ssadq_cfg.xml file contains the global configuration parameters for Oracle Data Quality Matching Server (IIR). To modify ssadq_cfg.xml, complete the steps in the following procedure. This task is a step in *Process of Installing the Oracle Data Quality Matching Server*. For an example ssadq_cfg.xml file, see *Sample Configuration Files for Oracle Data Quality Matching Server*.

## To modify configuration parameters for the Oracle Data Quality Matching Server

1. Open up a text editor.

2. Modify the following parameters in ssadq_cfg.xml, as required:

   a. Set `<iss_host>` to point to the server where Oracle Data Quality Matching Server is running.
   b. Set `<iss_port>` to 1666 (which is the default), unless you are using a different port for installation.
   c. Set the `<rulebase_name>` parameter. For example, with Oracle Database 10*g*:

      - username is ssa
      - password is SIEBEL
      - ServiceName is Target (as specified in the odbc.ini file for the Oracle Data Quality Matching Server server)
      - <rulebase_name> Example: odb:0:ssa/SIEBEL@Target

   d. Set <contact_system>, <account_system>, and <prospect_system> to the name of the system that you create in Oracle Data Quality Matching Server (IIR) using the SiebelDQ.sdf file.

      The system that you create in IIR (Console Client, Admin Mode) will hold all the IDT and IDX database tables.

      If you want to run Oracle Data Quality Matching Server against only a single entity (for example, Accounts) as opposed to multiple entities (Accounts, Contacts, and Prospects), then you must alter the definitions within the SiebelDQ.sdf file to include only the one entity that you want as otherwise the synchronizer fails to run. In this example, you must remove the definitions for Contacts and Prospects.

      Any changes that you make to the SDF file must be appended to the user property for the business service DQ Sync Services. If you do not want to use a particular field (for example, Birth Date) as part of deduplication, then that field must be removed from the SDF file. In addition, you must do the following:

      - Remove the corresponding mapping from data quality third-party administration settings in your Siebel application.
      - Change the user property in the DQ Sync Services business process. For example:

        For Account, change the Account_DeDupFlds user property.

        For Contact, change the Contact_DeDupFlds user property.
      - Remove the DeDup field from the user property.
      - Remove the corresponding mapping in the user property for external length.

        This is Account_ExtLen for Account, and Contact_ExtLen for Contact.

      Since CUT Address is shared across Account and Contact, any change in the CUT Address is reflected in both Account and Contact de-duplication.

3. Save the ssadq_cfg.xml file and copy to the SDQConnector folder on Siebel Server for changes to take effect:

   `siebsrvr/SDQConnector`

# Deploying Workflows for Oracle Data Quality Matching Server Integration

In the Siebel application, make sure that the following workflows are deployed and activated if using workflows to synchronize with the Oracle Data Quality Matching Server. This task is a step in *Process of Installing the Oracle Data Quality Matching Server*.

- ISS Build Load File
- ISS Delete Record Sync
- ISS ErrorHandler
- ISS Launch Build Load File
- ISS Launch Delete Record Sync
- ISS Launch PreDelete Record Sync
- ISS Launch PreWrite Record Sync
- ISS Launch Write Record Sync
- ISS PreDelete Record Sync
- ISS PreWrite Record Sync
- ISS WriteRecordNew
- ISS WriteRecordUpdated
- ISS Write Record Sync
- ISS Launch Record Sync
- ISS PreLaunch Record Sync
- ISS Launch Record ASync

These workflows are used to synchronize data between the Siebel application and Oracle Data Quality Matching Server (IIR) tables. For more information about Siebel Workflow, see *Siebel Business Process Framework: Workflow Guide* .

**Note:** The activation or deactivation of these workflows depends on your business needs. The business service, DQ Sync Services, can be used if you are using the multiple address feature.

# Initial Loading of Siebel Data into Oracle Data Quality Matching Server Tables

To initially load your Siebel application data into Oracle Data Quality Matching Server (IIR) tables, complete the steps in the following procedure. This procedure uses SQL scripts and is for large implementations where, for example, the database is too large to use an XML file import or export to initially load Siebel application data into Oracle Data Quality Matching Server tables.

**CAUTION:** Before proceeding any further, you must read, understand, and follow the following guidelines:

- It is highly recommended that data is directly loaded from source tables into Oracle Data Quality Matching Server tables.

  The sample system definition file (SiebelDQ.sdf) includes appropriate sections to load data directly from source tables into Oracle Data Quality Matching Server tables.

  > **Note:** For an example SDF file, see *Sample SiebelDQ.sdf File*.

- The system definition file includes information about the matching criteria for various entities.

  As part of the initial analysis, it is essential that you review the sample system definition file (SiebelDQ.sdf) and make appropriate changes to it, before creating any new systems in IIR.
- The sample system definition file (SiebelDQ.sdf) is not a preconfigured configuration file; it serves as a sample for you to start with.
- Make sure that the entries in the system definition file are in sync with the data quality configuration settings that you set up in your Siebel application (in Administration - Data Quality screen, Third Party Administration view).
- Make sure that the user properties that you set up in Siebel Tools for the business service are in sync with the entries in your system definition file.

> **Note:** If you encounter errors when trying to initially load a high volume of data (greater than 10,000 records), then set the system environment variable SSAOCI_IGNORE_UCS2_BYTES to one, and restart the Oracle Data Quality Matching Server server and client. Also, adding zeros when setting the SSA_XML_SIZE parameter can help when initially loading large files. For example: `set` SSA_XML_SIZE to 8000000.

This task is a step in *Process of Installing the Oracle Data Quality Matching Server*.

## To initially load Siebel application data into Oracle Data Quality Matching Server tables

1. Start the IIR Server by navigating to, for example, the following:

   Programs, Informatica, Identity Resolution, v2.8.07 (InformaticaIR), Informatica Identity Resolution, Informatica IR Server - Start(Configure Mode)
2. Start the IIR Console Client (in Admin Mode) by navigating to:

   Programs, Informatica, Identity Resolution, v2.8.07(InformaticaIR), Informatica Identity Resolution, Informatica IR Console Client - Start(Configure Mode)
3. If not already done so, create a new system in IIR using the appropriate System Definition file that you have reviewed and modified using the sample SiebelDQ.sdf file as a starting point. Or, if a system already exists, select it and refresh it by clicking the System/Refresh button.

   The system that you create in IIR (Console Client, Admin Mode) will hold all the IDT and IDX database tables.

   > **Note:** If you want to run IIR against only a single entity (for example, Accounts) as opposed to multiple entities (Accounts, Contacts, and Prospects), then you must alter the definitions within the SiebelDQ.sdf file to include only the one entity that you want as otherwise the synchronizer fails to run. In this example, you must remove the definitions for Contacts and Prospects.

4. Run the IDS_IDT_<ENTITY TO BE LOADED>_STG.sql script to take a snapshot of records in the Siebel application. For example, for account initial load, execute the following script from the SQL prompt as user SSA_SRC:

```
IDS_IDT_ACCOUNT_STG.sql
```

Depending on project requirements, IIR configuration, and data quality configuration, you must modify sample scripts provided with the software accordingly.

> **Note:** It is not mandatory to always load the data incrementally. If the initial volume of data to load is not high, then you can load the data directly from source tables to IIR tables in one go.

5. Run the IDS_IDT_CURRENT_BATCH_<ENTITY TO BE LOADED>.sql script to create the dynamic view to load the snapshot created in Step 4. For example, for account initial load, execute the following script from the SQL prompt as user SSA_SRC:

   ```
   IDS_IDT_CURRENT_BATCH_ACCOUNTS.sql
   ```

   To be in sync with the snapshot created in Step 4 and the SDF file used for system creation in Step 3, you must modify the sample scripts provided with the software according to project requirements, IIR configuration, and data quality configuration. Also, use a batch size that is appropriate to your project needs, initial data load volume, and any other project specific needs.
6. Run the following SQL script to create the database table to store the current batch number being loaded:

   ```
   IDS_IDT_CURRENT_BATCH.sql
   ```
7. Load IIR with data from the Siebel application by clicking the System/Load IDT button.
   Make sure to select the *All_load* option from the Loader Definition menu in the dialog that displays. This process loads records with batch number 1 from the snapshot created earlier. Validate the data to make sure that all the records with batch number 1 are correctly loaded.
8. Open a command window and navigate to the directory where the initial load scripts were copied during product installation.
9. Execute the initial load process by entering the following command at the command line:

   ```
   IDS_IDT_LOAD_ANY_ENTITY.CMD <Entity> <System> <Work Directory>
   ```

   For example, for account initial load, execute the following script:

   ```
   IDS_IDT_LOAD_ANY_ENTITY.CMD Account c:/initialLoad/logs
   ```

   This loads data in batches from the snapshot created in Step 4. The log files and error files recording the outcome of each batch load are stored in the `c:/InitialLoad/logs` directory.
10. Examine the log files and error files to identify any batch that failed to load. Use the information in the log and error files to determine the root cause for any failure and fix the underlying issue.
11. Incrementally load the failed batches individually using the following script from the command line:

    ```
    IDS_IDT_LOADBATCH_ANY_ENTITY.CMD
    ```

    For example, to load batch 33 of account, execute the following script from the command line:

    ```
    IDS_IDT_LOADBATCH_ANY_ENTITY.CMD Account SiebelDQ c:/initialLoad/logs 33
    ```
12. Examine the log files and error files to ensure that the (failed) batches successfully loaded. In case of errors, use the information in the log and error files to determine the root cause for the failure and fix the underlying issue. Repeat Step 11 until all the batches have successfully loaded.
13. Repeat this process to load other entities such as contacts and prospects.

# Upgrading to Informatica Identity Resolution 9.01

A new Cluster Id system field called CL ID is introduced by Informatica Identity Resolution 9.01. Consisting of 2 bytes, the CL ID field is added automatically to the end of each IDT table. To accommodate this change, you must configure a new CL ID field in DQ Sync Services so that Siebel can format the correct record length to insert into the Oracle Data Quality Matching Server NSA table.

Use the following procedure to upgrade your existing Oracle Data Quality Matching Server setup to Informatica Identity Resolution 9.01. This task is a step in *Configuring Oracle Data Quality Matching Server*.

## To upgrade to Informatica Identity Resolution 9.01

1. Add new CL ID fields to the DQ Sync Services business service user properties.

   > **Note:** Add the new CL ID field to the last field in your IDT table. For example, the last field in the Account IDT table is Account Business Address, so add the new CL ID field to the Account_Business_Address component (not to the parent Account object).

   - Add the following for Account:

| User Property | Value |
|---|---|
| Account Record Length | 1092 |
| Account_Business Address_ DataType | W\|W\|W\|W\|W\|C\|C |
| Account_Business Address_ DeDupFlds | City\|Country\|Postal Code\|State\|Street Address\|Address Id\|CL ID |
| Account_Business Address_ExtLen | 200\|120\|120\|40\|200\|60\|2 |

   - Add the following for Contact:

| User Property | Value |
|---|---|
| Contact Record Length | 1852 |
| Contact_INS Personal Address_ DataType | W\|W\|W\|W\|W\|C\|C |

| User Property | Value |
|---|---|
| Contact_INS Personal Address_ DeDupFld | INS Personal City\|INS Personal Country\|INS Personal Postal Code\|INS Personal State\|INS Personal Street Address\|INS Personal Address Id\|CL ID |
| Contact_INS Personal Address_ ExtLen | 120\|40\|40\|40\|200\|60\|2 |

- Add the following for Prospect:

| User Property | Value |
|---|---|
| Prospect Record Length | 1772 |
| Prospect_DataType | W\|W\|W\|W\|W\|W\|W\|W\|W\|W\|W\|W\|W\|C\|C |
| Prospect_DeDupFlds | Account\|Cellular Phone #\|City\|Country\|Email Address\|First Name Last Name\|Home Phone #\|Middle Name\|Postal Code\|Social Security Number\|State\|Street Address\|Work Phone #\|Id\|CL ID |
| Prospect_ExtLen | 200\|120\|120\|60\|120\|200\|120\|200\|40\|120\|40\|200\|200\|30\|2 |

2. Apply the Informatica Address Doctor Version 5 license in the ssaasmv5.xml file as follows:
   a. Open the ssaasmv5.xml file which is located in the `InformaticaIR/ssaas/ad5` folder.
   b. Configure the Informatica Address Doctor Version 5 license as follows:
      - Go to the `<ASM_ADv5_Config>` tag.
      - Set the unlock_code value as follows:

```
<ASM_ADv5_Config>
 <MAX_THREAD>1</MAX_THREAD>
 <MAX_ADOBJECTS>1</MAX_ADOBJECTS>
 <AD5_UNLOCK_CODE>
 <UNLOCK_CODE>unlock_code</UNLOCK_CODE>
 </AD5_UNLOCK_CODE>
</ASM_ADv5_Config>
```

      **Note:** If Informatica Identity Resolution 9.01 is being used on UNIX, the ssaasmv5.xml file has the following blank tag under <ASM_ADv5_CONFIG>, which must either be set with a proper value or completely removed from the ssaasmv5.xml file: <ENRICHMENT_OPTION> </ ENRICHMENT_OPTION>

   c. Save the ssaasmv5.xml file.

# Configuring Oracle Data Quality Matching Server

The Oracle Data Quality Matching Server provides real-time and batch data matching functionality using licensed third-party Informatica Identity Resolution software. For information about installing Oracle Data Quality Matching Server, see *Installing the Universal Connector*.

Configuring data quality for Oracle Data Quality Matching Server involves enabling deduplication on all object managers, specifying data quality settings, and setting up preconfigured vendor parameters and field mapping values for Oracle Data Quality Matching Server.

Use the following procedure to set up real-time deduplication for Oracle Data Quality Matching Server. This task is a step in *Configuring Oracle Data Quality Matching Server*.

## To configure data quality for Oracle Data Quality Matching Server

1. Enable data quality at the object manager level as described in *Enabling Data Quality at the Object Manager Level*.
2. Change the DeDuplication Data Type setting to ISS on all object managers as described in *Enabling Data Quality at the Enterprise Level*.

   This parameter can be set at the Enterprise, Siebel Server, or component level. For example, srvrmgr commands similar to the following can be used to set the parameters:

   ```
   Change param DeDupTypeType =ISS
   Change param DeDupTypeEnable =True, DeDupTypeType =ISS for comp DQMgr
   Change param DeDupTypeEnable =True, DeDupTypeType =ISS for comp SCCObjMgr_enu
   Change param DeDupTypeEnable =True, DeDupTypeType =ISS for comp UCMObjMgr_enu
   ```

   > **Note:** You must change the DeDuplication Data Type setting to ISS on all object managers for deduplication with Oracle Data Quality Matching Server to be active.

3. Set data quality settings as described in *Specifying Data Quality Settings*.

   Make sure that the following parameters are set to Yes:

   ```
   Enable DeDuplication
   Force User DeDupe - Account
   Force User DeDupe - Contact
   Force User DeDupe - List Mgmt
   ```

4. Verify that the preconfigured vendor parameter and field mapping values are set up as listed in *Universal Connector Parameter and Field Mapping Values for Oracle Data Quality Matching Server*.
5. Modify the ssadq_cfg.xml file as described in *Modifying Configuration Parameters for Oracle Data Quality Matching Server*

For more information about Siebel Server configuration and management, see *Siebel System Administration Guide* .

# Universal Connector Parameter and Field Mapping Values for Oracle Data Quality Matching Server

This topic includes information about the Universal Connector parameter and field mapping values for the Oracle Data Quality Matching Server:

- *Preconfigured Vendor Parameters for Oracle Data Quality Matching Server*
- *Preconfigured Field Mappings for Oracle Data Quality Matching Server*

## Preconfigured Vendor Parameters for Oracle Data Quality Matching Server

The following table lists the vendor parameters preconfigured for Oracle Data Quality Matching Server (Informatica Identity Resolution). You must not reconfigure the parameter settings.

| Name | Value |
|---|---|
| Account DeDup Record Type | Account |
| Contact DeDup Record Type | Contact |
| List Mgmt Prospective Contact DeDup Record Type | Prospect |
| Parameter 1 | "global", "iss-config-file", "ssadq_cfg.xml" |
| Candidate Acquisition Process by Third Party | Yes.<br><br>If set to Yes, then match candidate acquisition takes place using the Oracle Data Quality Matching Server.<br><br>The default value for this parameter is No. |

## Preconfigured Field Mappings for Oracle Data Quality Matching Server

This topic includes information about the preconfigured Oracle Data Quality Matching Server (Informatica Identity Resolution) field mappings for the following business components:

- *Preconfigured Field Mappings for Business Component - Account*
- *Preconfigured Field Mappings for Business Component - Contact*

- *Preconfigured Field Mappings for Business Component - List Mgmt Prospective Contact*
- *Field Mappings for Business Component - CUT Address*

## Preconfigured Field Mappings for Business Component - Account

The following table shows the Oracle Data Quality Matching Server data matching field mappings for the Account business component and DeDuplication operation.

| Business Component Field | Mapped Field |
|---|---|
| DUNS Number | DUNSNumber |
| Name | Name |
| Primary Account City | PAccountCity |
| Primary Account Country | PAccountCountry |
| Primary Account Postal Code | PAccountPostalCode |
| Primary Account State | PAccountState |
| Primary Account Street Address | PAccountStrAddress |
| Row Id | RowId |

## Preconfigured Field Mappings for Business Component - Contact

The following table shows the Oracle Data Quality Matching Server data matching field mappings for the Contact business component and DeDuplication operation.

| Business Component Field | Mapped Field |
|---|---|
| Birth Date | BirthDate |
| Cellular Phone # | CellularPhone |
| Email Address | EmailAddress |
| First Name Last Name | NAME |
| Home Phone # | HomePhone |
| Middle Name | MiddleName |

| Business Component Field | Mapped Field |
|---|---|
| Primary Account Name | Account |
| Primary Personal City | City |
| Primary Personal Country | Country |
| Primary Personal Postal Code | PrimaryPostalCode |
| Primary Personal State | State |
| Primary Personal Street Address | StreetAddress |
| Row Id | RowId |
| Social Security Number | SocialSecurityNumber |
| Work Phone # | Work Phone |

## Preconfigured Field Mappings for Business Component - List Mgmt Prospective Contact

The following table shows the Oracle Data Quality Matching Server data matching field mappings for the List Mgmt Prospective Contact business component and DeDuplication operation.

| Business Component Field | Mapped Field |
|---|---|
| Account | Account |
| Cellular Phone # | CellularPhone |
| City | City |
| Country | Country |
| Email Address | EmailAddress |
| First Name Last Name | NAME |
| Home Phone # | HomePhone |
| Id | RowId |

| Business Component Field | Mapped Field |
|---|---|
| | |
| Middle Name | MiddleName |
| Postal Code | PostalCode |
| Social Security Number | SocialSecurityNumber |
| State | State |
| Street Address | StreetAddress |
| Work Phone | WorkPhone |

## Field Mappings for Business Component - CUT Address

The following table shows the data matching field mappings for the CUT Address business component and DeDuplication operation. In your Siebel application, you must manually add these field mappings if you want to configure deduplication against multiple addresses, which is supported only by Oracle Data Quality Matching Server. For more information, see *Configuring Deduplication Against Multiple Addresses*.

**Note:** For Siebel Industry Applications, the CUT Address business component is used instead of the Business Address business component.

| Business Component Field | Mapped Field |
|---|---|
| City | PAccountCity |
| Country | PAccountCountry |
| Postal Code | PAccountPostalCode |
| Row Id | PAccountAddressID |
| State | PAccountState |
| Street Address | PAccountStrAddress |

# Configuring a New Field for Real-Time Data Matching

When using the Oracle Data Quality Matching Server for data matching, there are a number of steps involved in configuring a new field for real-time data matching. Use the following procedure to configure a new field for data matching. This procedure shows you how to add a new field called *Position* when matching Contacts.

## To configure a new field for real-time data matching

1. In your Siebel application, navigate to the Administration - Data Quality screen, then the Third Party Administration view, and add the vendor field mappings for the new business components. For example:

   a. Select the Vendor shown in the following table:

| Vendor Name | DLL Name |
|---|---|
| ISS | ssadqsea |

   b. Select the business component operation shown in the following table:

| Business Component Name | Operation |
|---|---|
| Contact | DeDuplication |

   c. Add the field mapping shown in the following table:

| Business Component Field | Mapped Field |
|---|---|
| Position | MyPosition |

2. Configure the integration object.

   a. If using the old synchronizer, modify the Identity Search Server synchronization Integration Object by adding the new fields to it. In this example, you must modify the ISS_Contact to add the new Integration Component Field as shown in the following table:

| Name | Data Type | Length | External Name | XML Tag |
|---|---|---|---|---|
| Position | DTYPE_TEXT | 50 | Position | MyPosition |

| Name | Data Type | Length | External Name | XML Tag |
|---|---|---|---|---|
| | | | | |

> **Note:** For a Contact, you must modify the ISS_Contact integration object. For an Account, you must modify the ISS_Account integration object. For a Prospect, you must modify the ISS_List_Mgmt_Prospective_Contact integration object.

b. If using the new synchronizer, then do the following:

- Add the new field to the Synchronize Integration Object. In this case, 'SyncContact' IO. For the contact address field, add it to the 'Contact_INS Personal Address' Integration Component.
- Add the new field to the 'DQ Sync Services' user property. In this case, 'Contact_DeDupFlds'. For the contact address field, add it to the 'Contact_INS Personal Address_DeDupFlds' user property.
- Add the new field length to one or both of the following 'DQ Sync Services' user properties:

```
'Contact_ExtLen'
'Contact_INS Personal Address_ExtLen'
```

> **Note:** It is mandatory that you maintain the same sequence that is detailed in the sdf file. Also make sure that the address fields are grouped together at the end of the sdf file.

  - Enter the new record length into the 'Contact Record Length' user property. This user property holds a total of all the field lengths in 'Contact_ExtLen' and 'Contact_INS Personal Address_ExtLen'.

> **Note:** ISSDataSrc must be added to the OM - Named Data Source component parameter for the UCM object manager and EAI object manager components.

3. Modify the SDF file (SiebelDQ.sdf):

a. Add the new fields to the IDT table in IIR:

In the following example for the old synchronizer, you must add the new Position field to the IDT_Contact. For example:

```
create_idt
 IDT_CONTACT
SOURCED_FROM FLAT_FILE
 BirthDate w(60),
 CellularPhone w(60),
 EmailAddress w(60),
 NAME w(100),
 HomePhone w(60),
 MiddleName w(100),
 Account w(100),
 City w(60),
 Country w(20),
 PrimaryPostalCode w(20),
 State w(20),
 StreetAddress w(100),
 RowId w(30)
 SocialSecurityNumber w(60)
 MyPosition w(60)
 WorkPhone w(60)
SYNC_REPLACE_DUPLICATE_PK
TXN-SOURCE NSA
```

```
;
```

In the following example for the new synchronizer, you must add the new Position field to the IDT_Contact.

```
create_idt IDT_CONTACT
sourced_from odb:15:ssa_src/ssa_src@ISS_DSN
INIT_LOAD_ALL_CONTACTS.BIRTHDATE BirthDate V(60),
INIT_LOAD_ALL_CONTACTS.CELLULARPHONE CellularPhone V(60),
INIT_LOAD_ALL_CONTACTS.EMAILADDRESS EmailAddress V(60),
INIT_LOAD_ALL_CONTACTS.NAME NAME V(100),
INIT_LOAD_ALL_CONTACTS.HOMEPHONE HomePhone V(60),
INIT_LOAD_ALL_CONTACTS.MIDDLE_NAME MiddleName V(100),
INIT_LOAD_ALL_CONTACTS.ACCOUNT Account V(100),
INIT_LOAD_ALL_CONTACTS.CONTACT_ID (pk1) RowId C(30),
INIT_LOAD_ALL_CONTACTS.SOCIALSECURITYNUMBER SocialSecurityNumber V(60),
INIT_LOAD_ALL_CONTACTS.MYPOSITION MyPosition V(60),
INIT_LOAD_ALL_CONTACTS.WORKPHONE WorkPhone V(60),
INIT_LOAD_ALL_CONTACTS.CITY City V(60),
INIT_LOAD_ALL_CONTACTS.COUNTRY Country V(20),
INIT_LOAD_ALL_CONTACTS.POSTAL_CODE PrimaryPostalCode V(20),
INIT_LOAD_ALL_CONTACTS.STATE State V(20),
INIT_LOAD_ALL_CONTACTS.STREETADDRESS StreetAddress V(100),
INIT_LOAD_ALL_CONTACTS.ADDRESS_ID (pk2) ContactAddressID C(60)
SYNC REPLACE_DUPLICATE_PK
TXN-SOURCE NSA
;
```

**b.** Modify SCORE-LOGIC in the IIR search definition:

A set of field types are provided that are supported by Match Purpose. For Contact Match Purpose, the required and optional field types are shown in the following table:

| Field | Required |
|---|---|
| Person_Name | Yes |
| Organization_Name | Yes |
| Address_Part1 | Yes |
| Address_Part2 | No |
| Posal_Area | No |
| Telephone_Number | No |
| ID | No |
| Date | No |

| Field | Required |
|---|---|
| | |
| Attribute1 | No |
| Attribute2 | No |

If you want the new field to contribute to the match score, add it to the SCORE-LOGIC section in IIR search definition. For example:

```
search-definition
==================
NAME= "search-person-name"
IDX= IDX_CONTACT_NAME
COMMENT= "Use this to search and score on person"
KEY-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("FIELD=Person_Name
 SEARCH_LEVEL=Typical UNICODE_ENCODING=6",
 Field(Name)
SCORE-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("Purpose=Person_Name
 MATCH_LEVEL=Typical UNICODE_ENCODING=6",
 Matching-Fields
("Name:Person_Name,StreetAddress:Address_Part1,City:Address_part2,State:Attr
ibute1,PrimaryPostalCode:Postal_area,MyPosition:Attribute2")
```

4. Delete the existing system in IIR, and then create a new system using the new SiebelDQ.sdf file.
5. Reload the IIR system as described in *Initial Loading of Siebel Data into Oracle Data Quality Matching Server Tables*.

# Loading Data Incrementally

When loading Siebel application data into Informatica Identity Resolution tables, rather than generating keys for the entire data set in one go, the incremental data load feature provides the functionality to divide the total load into smaller batches of configurable size and load them incrementally. Use the following procedure to configure a new field for data matching.

## To incrementally load Siebel application data into Informatica Identity Resolution tables

1. Start the Informatica Identity Resolution server in Configure Mode.
2. Start the Informatica Identity Resolution Console client in Admin Mode.
3. Select System (if already created), otherwise create a new system using the required sdf file.
4. Run the IDS_IDT_<ENTITY TO BE LOADED>_STG.sql script to take a snapshot of records in the Siebel application.

For example, execute the following script from the SQL command line prompt for account initial load using your appropriate login credentials:

```
IDS_IDT_ACCOUNT_STG.sql
```

The following sample SQL scripts can be used to capture snapshots of the data:

- IDS_IDT_ACCOUNT_STG.SQL
- IDS_IDT_CONTACT_STG.SQL
- IDS_IDT_PROSPECT_STG.SQL

For more information about these example scripts, see *Sample SQL Scripts*.

5. While creating a snapshot using the example scripts listed in the previous step, users are prompted to enter a batch size. Depending on the value entered, the entire snapshot is grouped into batches of the specified batch size.

   For example, run the following SQL script to create the database table to store the current batch number being loaded (this value is usually 1 for the first time):

   ```
   IDS_IDT_CURRENT_BATCH.sql
   ```

6. Run the IDS_IDT_CURRENT_BATCH_<ENTITY TO BE LOADED>.sql script to create the dynamic view to load the snapshot for the staging table created in the previous step.

   For example, execute the following script from the command line:

   ```
   IDS_IDT_CURRENT_BATCH_ACCOUNT.sql
   ```

   The following sample SQL scripts can be used to create the views to process the records in a given batch:

   - IDS_IDT_CURRENT_BATCH_ACCOUNT.SQL
   - IDS_IDT_CURRENT_BATCH_CONTACT.SQL
   - IDS_IDT_CURRENT_BATCH_PROSPECT.SQL

   For more information about these example scripts, see *Sample SQL Scripts*.

7. Open the Informatica Identity Resolution client and perform a Load IDT.

   Load the remaining batches of data through the ISS batch Utility. Open a command window and navigate to the directory where the initial scripts for loading have been copied. Execute the initial load process by entering the following command at the command line:

   ```
   IDS_IDT_LOAD_ANY_ENTITY.CMD <Entity> <Work Directory>
   ```

   For example, execute the following script from the command line for account initial load:

   ```
   IDS_IDT_LOAD_ANY_ENTITY.CMD Account c:/initialLoad/logs
   ```

8. The following files contain the parameters used by the batch load utility; you must update these files to reflect your installation:
    - idt_Account_load.txt
    - idt_Contact_load.txt
    - idt_Prospect_load.txt

    **Note:** Certain SQL and shell scripts are required to create materialized views and to load data incrementally. Depending on customer requirements, you can fine tune these sample files during implementation.

9. Incrementally load the failed batches individually using the following script from the command line:

    ```
    IDS_IDT_LOADBATCH_ANY_ENTITY <Entity> <Work Directory> <Batch Number>
    ```

    For example, execute the following script from the command line to load batch 33 of account:

    ```
    IDS_IDT_LOADBATCH_ANY_ENTITY.CMD Account c:/initialLoad/logs 33
    ```

    Examine the log files and error files to ensure that all batches have successfully loaded. In the case of errors, use the information in the log and error files to determine the root cause for the failure and fix the underlying issue; repeat the load process as necessary.

# Process of Configuring Data Synchronization Between Siebel CRM and Oracle Data Quality Matching Server

When configuring data synchronization between the Siebel application and Oracle Data Quality Matching Server, perform the following tasks:

1. Configure the data source, see *Configuring the Data Source*.
2. Configure the DQ Sync Services Business Service to Use Informatica Identity Resolution 9.01, see *Upgrading to Informatica Identity Resolution 9.01* (*Upgrading to Informatica Identity Resolution 9.01*).
3. *Configuring Object Managers for ISS Data Source*.

## Configuring the Data Source

Use the following procedure to configure the data source definition (or EBC table) required to synchronize data between the Siebel application and Oracle Data Quality Matching Server. This task is a step in *Process of Configuring Data Synchronization Between Siebel CRM and Oracle Data Quality Matching Server*.

### To configure the data source definition

1. In your Siebel application, navigate to Administration - Server Configuration, Enterprises, and then Profile Configuration view.
2. Copy an existing InfraDatasources named subsystem type.
3. Change the Profile and Alias properties to the Data Source name (ISSDataSrc) configured in Web Tools.
4. Update the profile parameters to correspond to the external RDBMS:
    - DSConnectString: This is the data source connect string.

For the Microsoft SQL Server or the IBM DB2 databases, create an ODBC or equivalent connection and input the name of the connection in the parameter. For an Oracle RDBMS, specify the TNS name associated with the database, and not an ODBC or other entry.

- DSSSQLStyle: This is the database SQL type.
- DSDLLName: This is the DLL Name corresponding to the SQL type.
- DSTableOwner: This is the data source table owner.
- DSUsername: This is the default user name used for connections (Optional).
- DSPassword: This is the default password used for connections (Optional).

# Configuring Object Managers for ISS Data Source

Use the following procedure to configure object managers for the ISS Data Source. This task is a step in *Process of Configuring Data Synchronization Between Siebel CRM and Oracle Data Quality Matching Server*.

## To configure object managers for ISS Data Source

1. Navigate to Administration - Server Configuration, Enterprises, and then Component Definitions view.
2. Query for the components used with UCM, such as, the following:
   - EAI
   - UCM
   - Data Quality
   - Workflow
3. Select a component, and then select Start Reconfiguration from the main menu.
4. In the list applet that appears after the Component Definitions applet, query for the following parameter: OM - Named Data Source.

   **Note:** The value for this parameter is similar to the following: `ServerDataSrc,GatewayDataSrc`.

5. Add a comma after the last data source, then add the ISS data source you created in *Configuring the Data Source*. The default data source name is ISSDataSrc.
6. Save the record, then select Commit Reconfiguration from the main menu.
7. Repeat Step 3 through Step 6 for all required Object Managers.

   For example, add ISSDataSrc to the following components:

   - EAI Object Manager: OM - Named Data Source name
   - UCM Object Manager: OM - Named Data Source name
   - UCM Batch Manager: OM - Named Data Source name
   - Data Quality Manager
   - Workflow Process Batch Manager
   - Workflow Process Manager
8. Click Synchronize to save the changes to the Siebel Gateway.
9. Start the Gateway Server and the Siebel Server.

# Siebel CRM DQ Sync Action Sets

DQ Sync action sets use the Siebel application business service to insert records directly into Oracle Data Quality Matching Server database tables. Activate DQ Sync action sets. For information about creating action sets, including creating actions for action sets, and associating events with action sets, see *Siebel Personalization Administration Guide* .

This topic introduces the following DQ Sync action sets for Siebel applications:

- *DQ Sync Action Sets for Account*
- *DQ Sync Action Sets for Contact*
- *DQ Sync Action Sets for List Mgmt Prospective Contact*
- *Generic DQ Sync Action Sets*

## DQ Sync Action Sets for Account

This topic introduces the following Siebel application DQ Sync action sets for Account:

- *DQ Sync DeleteRecord Account*
- *DQ Sync PreDeleteRecord Account*
- *DQ Sync PreWriteRecord Account*
- *DQ Sync WriteRecord Account*

For more information about creating action sets, including creating actions for action sets, and associating events with action sets, see *Siebel Personalization Administration Guide* .

### DQ Sync DeleteRecord Account

The following table describes the actions in the DQ Sync DeleteRecord Account action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | siebeldq |
| ISS Set IDT Name | Name | ISS Set IDT Name |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_ACCOUNT |
| ISS Set BC Name | Name | ISS Set BC Name |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_BC_NAME |
| | Set Operator | Set |
| | Value | Account |
| ISS Run DQSync | Name | ISS Run DQSync |
| | Sequence | 4 |
| | Action Type | BusService |
| | Business Service Name | DQ Sync Services |
| | Business Service Method | SyncISS |

## DQ Sync PreDeleteRecord Account

The following table describes the actions in the DQ Sync PreDeleteRecord Account action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |

| Action | Name of Field | Value |
|---|---|---|
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | siebeldq |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_ACCOUNT |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Set BC Name | Name | ISS Set BC Name |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_BC_NAME |

| Action | Name of Field | Value |
|---|---|---|
| | Set Operator | Set |
| | Value | Account |
| ISS Run DQSync | Name | ISS Run DQSync |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | DQ Sync Services |
| | Business Service Method | SyncISS |

## DQ Sync PreWriteRecord Account

The following table describes the actions in the DQ Sync PreWriteRecord Account action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | siebeldq |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Value | IDS_01_IDT_ACCOUNT |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Set BC Name | Name | ISS Set BC Name |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_BC_NAME |
| | Set Operator | Set |
| | Value | Account |
| ISS Run DQSync | Name | ISS Run DQSync |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | DQ Sync Services |
| | Business Service Method | SyncISS |

## DQ Sync WriteRecord Account

The following table describes the actions in the DQ Sync WriteRecord Account action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | siebeldq |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_ACCOUNT |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Set BC Name | Name | ISS Set BC Name |
| | Sequence | 4 |
| | Action Type | Attribute Set |

| Action | Name of Field | Value |
|---|---|---|
| | Profile Attribute | IDS_BC_NAME |
| | Set Operator | Set |
| | Value | Account |
| ISS Run DQSync | Name | ISS Run DQSync |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | DQ Sync Services |
| | Business Service Method | SyncISS |

# DQ Sync Action Sets for Contact

This topic introduces the following Siebel application DQ Sync action sets for Contact:

- *DQ Sync DeleteRecord Contact*
- *DQ Sync PreDeleteRecord Contact*
- *DQ Sync PreWriteRecord Contact*
- *DQ Sync WriteRecord Contact*

## DQ Sync DeleteRecord Contact

The following table describes the actions in the DQ Sync DeleteRecord Contact action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |

| Action | Name of Field | Value |
|---|---|---|
| | Value | siebeldq |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_CONTACT |
| ISS Set BC Name | Name | ISS Set BC Name |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_BC_NAME |
| | Set Operator | Set |
| | Value | Contact |
| ISS Run DQSync | Name | ISS Run DQSync |
| | Sequence | 4 |
| | Action Type | BusService |
| | Business Service Name | DQ Sync Services |
| | Business Service Method | SyncISS |

## DQ Sync PreDeleteRecord Contact

The following table describes the actions in the DQ Sync PreDeleteRecord Contact action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | siebeldq |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_CONTACT |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Set BC Name | Name | ISS Set BC Name |
| | Sequence | 4 |
| | Action Type | Attribute Set |

| Action | Name of Field | Value |
|---|---|---|
| | Profile Attribute | IDS_BC_NAME |
| | Set Operator | Set |
| | Value | Contact |
| ISS Run DQSync | Name | ISS Run DQSync |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | DQ Sync Services |
| | Business Service Method | SyncISS |

## DQ Sync PreWriteRecord Contact

The following table describes the actions in the DQ Sync PreWriteRecord Contact action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | siebeldq |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Set Operator | Set |
| | Value | IDS_01_IDT_CONTACT |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Set BC Name | Name | ISS Set BC Name |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_BC_NAME |
| | Set Operator | Set |
| | Value | Contact |
| ISS Run DQSync | Name | ISS Run DQSync |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | DQ Sync Services |
| | Business Service Method | SyncISS |

## DQ Sync WriteRecord Contact

The following table describes the actions in the DQ Sync WriteRecord Contact action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | siebeldq |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_CONTACT |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Set BC Name | Name | ISS Set BC Name |

| Action | Name of Field | Value |
|---|---|---|
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_BC_NAME |
| | Set Operator | Set |
| | Value | Contact |
| ISS Run DQSync | Name | ISS Run DQSync |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | DQ Sync Services |
| | Business Service Method | SyncISS |

# DQ Sync Action Sets for List Mgmt Prospective Contact

This topic introduces the following Siebel application DQ Sync action sets for List Mgmt Prospective Contact:

- *DQ Sync DeleteRecord Prospect*
- *DQ Sync PreDeleteRecord Prospect*
- *DQ Sync PreWriteRecord Prospect*
- *DQ Sync WriteRecord Prospect*

## DQ Sync DeleteRecord Prospect

The following table describes the actions in the DQ Sync DeleteRecord Prospect action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |

| Action | Name of Field | Value |
|---|---|---|
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | siebeldq |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_PROSPECT |
| ISS Set BC Name | Name | ISS Set BC Name |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_BC_NAME |
| | Set Operator | Set |
| | Value | Prospect |
| ISS Run DQSync | Name | ISS Run DQSync |
| | Sequence | 4 |
| | Action Type | BusService |
| | Business Service Name | DQ Sync Services |
| | Business Service Method | SyncISS |

## DQ Sync PreDeleteRecord Prospect

The following table describes the actions in the DQ Sync PreDeleteRecord Prospect action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | siebeldq |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_PROSPECT |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Set BC Name | Name | ISS Set BC Name |

| Action | Name of Field | Value |
|---|---|---|
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_BC_NAME |
| | Set Operator | Set |
| | Value | Prospect |
| ISS Run DQSync | Name | ISS Run DQSync |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | DQ Sync Services |
| | Business Service Method | SyncISS |

## DQ Sync PreWriteRecord Prospect

The following table describes the actions in the DQ Sync PreWriteRecord Prospect action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | siebeldq |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_PROSPECT |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Set BC Name | Name | ISS Set BC Name |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_BC_NAME |
| | Set Operator | Set |
| | Value | Prospect |
| ISS Run DQSync | Name | ISS Run DQSync |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | DQ Sync Services |
| | Business Service Method | SyncISS |

| Action | Name of Field | Value |
|---|---|---|
| | | |

## DQ Sync WriteRecord Prospect

The following table describes the actions in the DQ Sync WriteRecord Prospect action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | siebeldq |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_PROSPECT |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |

| Action | Name of Field | Value |
|---|---|---|
| | Value | [Id] |
| ISS Set BC Name | Name | ISS Set BC Name |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_BC_NAME |
| | Set Operator | Set |
| | Value | Prospect |
| ISS Run DQSync | Name | ISS Run DQSync |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | DQ Sync Services |
| | Business Service Method | SyncISS |

# Generic DQ Sync Action Sets

This topic introduces the following Siebel application generic DQ Sync action sets for all business components:

- *DQ Sync UpdateAddress*
- *DQ Sync WriteRecordNew*
- *DQ Sync WriteRecordUpdated*

## DQ Sync UpdateAddress

The following table describes the actions in the DQ Sync UpdateAddress action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Run DQSync | Name | ISS Run DQSync |
| | Sequence | 1 |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Action Type | BusService |
| | Business Service Name | DQ Sync Services |
| | Business Service Method | SyncISS |

## DQ Sync WriteRecordNew

The following table describes the actions in the DQ Sync WriteRecordNew action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS UPDATE FLAG | Name | ISS UPDATE FLAG |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IS_UPDATE |
| | Set Operator | Set |
| | Value | false |

## DQ Sync WriteRecordUpdated

The following table describes the actions is in the DQ Sync WriteRecordUpdated action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS UPDATE FLAG | Name | ISS UPDATE FLAG |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IS_UPDATE |
| | Set Operator | Set |

| Action | Name of Field | Value |
|---|---|---|
| | Value | true |

# Configuring Deduplication Against Multiple Addresses

When using the Oracle Data Quality Matching Server for data matching, you can configure deduplication against either the primary address or all address entities. When Oracle Data Quality is configured to carry out deduplication against all address entities, this helps identify duplicates between records that have similar non primary addresses. Previous to this release, deduplication was carried out on the primary address only.

- For account, you must use the Account Match Against parameter to specify whether to match using one of the following:
  - All Address
  - Primary Address
- For contact, you must use the Contact Match Against parameter to specify whether to match using one of the following:
  - All Address
  - Primary Address

**Note:** You cannot perform deduplication against both All Address and Primary Address. Only one option can be used for deduplication. Choosing to carry out deduplication against all addresses is performance intensive.

The following procedure describes how to configure deduplication against multiple addresses. Once configured, deduplication against multiple addresses applies in real-time, Universal Customer Master (UCM) or Enterprise Application Integration (EAI) insertion, and batch match processing modes.

## To configure deduplication against multiple addresses

1. In your Siebel application, navigate to the Administration - Data Quality screen, then the Third Party Administration view, select the ISS Vendor Name (DLL Name: ssadqsea) and:
   a. In the BC Vendor Field Mapping, configure the business component operation shown in the following table:

| Business Component Field | Operation |
|---|---|
| CUT Address | DeDuplication |

| Business Component Field | Operation |
|---|---|
| | |

b. In the Vendor Parameter, configure the value shown in the following table:

| Name | Value |
|---|---|
| CUT Address DeDup Record Type | CUT Address |

c. In the field mapping for CUT Address, enter the values shown in the following table:

| Business Component Field | Mapped Field |
|---|---|
| PositionCity | PAccountCity |
| Country | PAccountCountry |
| Postal Code | PAccountPostalCode |
| Row Id | PAccountAddressID |
| State | PAccountState |
| Street Address | AccountStrAddress |

2. In your Siebel application, navigate to the Administration - Data Quality screen, then the Data Quality Settings view, and:

a. In the Value field for the parameters shown in the following table, specify the appropriate settings.
b. Log out of the application and log back in for the changes to take effect (you do not have to restart the Siebel Server).

| Parameter | Description |
|---|---|
| Account Match Against | Set to one of the following:<br>- All Address to consider all addresses associated with an account for deduplication.<br>- Primary Address (the default value). If set to Primary Address, then only the primary address associated with an account is considered for deduplication. |
| Contact Match Against | Set to one of the following: |

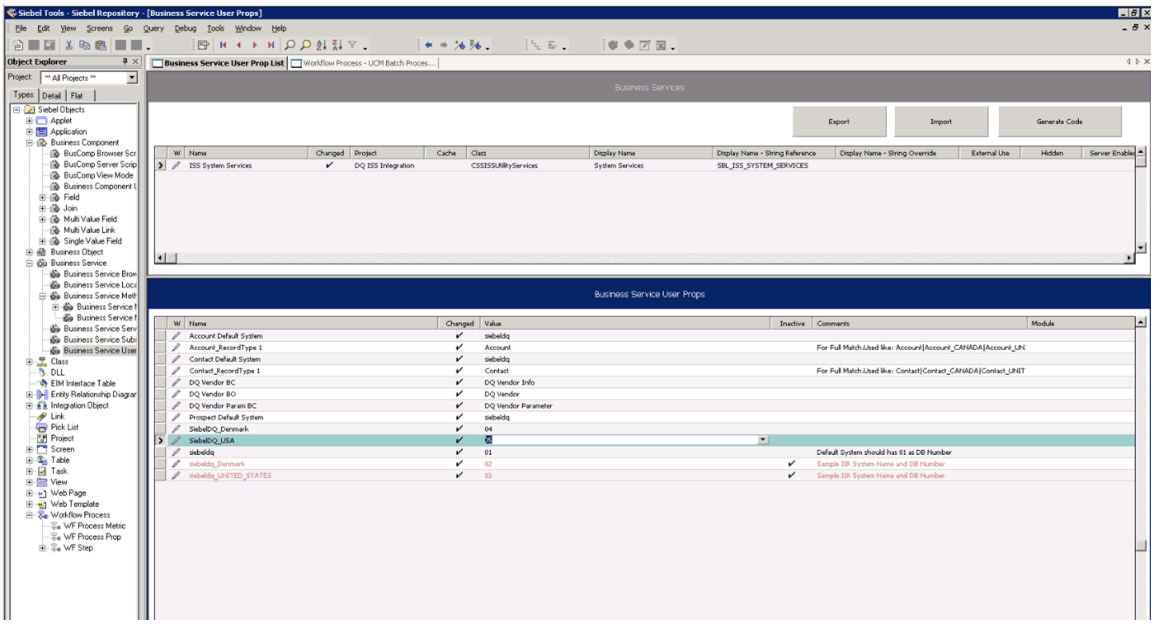
| Parameter | Description |
|---|---|
| | - All Address to consider all addresses associated with a contact for deduplication.<br>- Primary Address (default value). If set to Primary Address, then only the primary address associated with a contact is considered for deduplication. |
| Enable DQ Multiple Addresses | Set the value of this property Yes. |
| Enable DQ Sync | Set the value of this property to Yes. |

# Configuring Multiple Language Support for Data Matching

When using Oracle Data Quality Matching Server for data matching, you can configure different match rules for different languages. This is useful when you have multiple geographical implementations, and for each implementation, you want to use country-specific match rules.

The solution involves creating multiple systems in Informatica Identity Resolution, where each system corresponds to a specific country or language. Records with a specific language or country are routed to the corresponding system. Because each system is linked to a population, the respective country-specific population rules are used for matching the records. To use this feature, add the *Append BC Record Type Field x* vendor parameter for UI entry and the *Batch Append BC Record Type Field x* vendor parameter for batch mode. This vendor parameters is used to specify a field in the BC, which has the country or language information. The BC name can be Account, Contact, or Prospect where *x* represents the sequence number. For example: `Batch Append Account Record Type Field 1.`

You must add similar vendor parameters to the Business Service User Property of the ISS System Services business service to represent the IDT number corresponding to the Language. When you create multiple systems in Informatica Identity Resolution, you must specify a unique database number for each system which is used as part of the unique IDT table name in Informatica Identity Resolution. You must enter this same number into the Business Service User Property of the ISS System Services business service for each system. The name of the user property is the same as the name of the system created in Informatica Identity Resolution, and the value of the user property is the database number (for example, see the following figure).




**Note:** In order to pick up all the records that belong to the same country when running a data quality batch processing task, it is mandatory to define a search specification (to pick up the records belonging to the same country). You can define a search specification by navigating to the Administration - Server Management screen, then the Jobs view.

This feature can be extended as follows:

- Extended to have different match rules depending on the source of data (for example the Siebel application or other application).
- Extended to have different match rules depending on the mode of data entry (for example, real-time or batch processing mode). The procedure below describes how to configure multiple mode support for data matching when using the Oracle Data Quality Matching Server for data matching.

Use the following procedure to configure multiple language support for data matching when using the Oracle Data Quality Matching Server for data matching.

## To configure the Siebel Server for Informatica Identity Resolution multiple language support

1. Create systems on the Informatica Identity Resolution server.

    a. Create separate SDF files for each Country (Population). Informatica provides Standard Populations for most of the countries. Standard Populations are distributed as part of SSA-NAME3 installation and can be copied separately if not selected when installing NAME3 server.
    b. Once all populations are in place, check and note the filename of each population, as this is the same name that is used in the SDF file.

    You can change System Name and System ID within the system definition file as follows:

```
system-definition
*=================
NAME= siebeldq_XXXX
```

```
ID= sYY
```

Replace XXXX with Country, and YY with any number between and including 02 and 99. System ID 01 is reserved for Default System. For example, for Japanese population:

```
filename : siebeldq_Japan.sdf
Population files :
japan.ysp
```

Changes to SDF file:

```
system-definition
*=================
NAME= siebeldq_Japan
ID= s05
```

System ID 01 is reserved for Default System

Replace all occurrences of Population(usa) to Population(japan).

Similar changes are required for each sdf file.

2. Configure the ssadq_cfg.xml configuration file in the `<siebsrvr>/SDQConnector` folder.

   For example, add the following parameter to the ssadq_cfg.xml file:

```
<Parameter>
 <Record_Type>
 <Name>Account_Japan</Name>
 <System>siebeldq_Japan</System>
 <Search>search-org</Search>
 <no_of_sessions>25</no_of_sessions>
 </Record_Type>
</Parameter>
```

3. In your Siebel application, navigate to the Administration - Data Quality screen, then the Data Quality Settings view and in the value field for the parameter shown in the following table, specify the following setting:

| Parameter Name | Value |
|---|---|
| Enable DQ Multiple Languages | Yes |

4. Enable multiple language support for real-time flow:

   a. Navigate to Administration - Data Quality screen, then Third Party Administration view in your Siebel application.
   b. Select ISS as the third party vendor.
   c. Add the vendor parameters shown in the following table:

| Name | Value |
|---|---|
| Append Account Record Type Field 1 | Primary Account Country |

| Name | Value |
|---|---|
| Append Contact Record Type Field 1 | Primary Personal Country |

5. Enable multiple language support for batch flow:
   a. Navigate to Administration - Data Quality screen, then Third Party Administration view in your Siebel application.
   b. Select ISS as the third-party vendor.
   c. Add the vendor parameters shown in the following table:

| Name | Value |
|---|---|
| Batch Append Account Record Type Field 1 | Country |
| Batch Append Contact Record Type Field 1 | Primary Personal Country |

6. Add the user property to the ISS System Services business service.

   The user property that you add must correspond to the system name created in Informatica Identity Resolution for the respective country. For example, if the system created for Japan is siebeldq_Japan and the ID is set to 5, then the user property name must be siebeldq_Japan and the value 05, as shown in the following table.

| User Property Name | Value |
|---|---|
| siebeldq_Japan | 05 |

# Configuring Multiple Mode Support for Data Matching

When using Oracle Data Quality Matching Server for data matching, you can configure different match rules for data matching depending on:

- Mode of operation (real-time or batch mode)
- Source of data (Siebel CRM or other application, such as, eBiz)

As a prerequisite to configuring multiple mode support for data matching, multiple language support for data matching must be configured as described in *Configuring Multiple Language Support for Data Matching*.

Follow the steps in the following procedure in order to use different match rules on a custom parameter (*Source System*). Using this procedure, the match rules that apply to data from source system 1 (EBIZ) will be different to the match rules that apply to data from source system 2 (SIEBEL).

# To configure multiple mode support for data matching

1. Define separate match rules for each source system.
2. Create systems corresponding to each source system, where each source system points to a separate SDF file. For example:
   - Siebeldq_ebiz
   - Siebeldq_siebel

   The SDF file contains the IDT Layout Definition, Key Definition Logic, Match Fields considered for scoring records, the population to be used, and the Match purpose.

   **Note:** You must contact Informatica Technical Support in order to fine tune the SDF file.
3. Apply changes to the ssadq_cfg.xml file.

   For each system that you create in IIR, add the following parameters. There must be two entries, one for each source system (EBIZ and SIEBEL).

```
<Record_Type>
<Name>BCNAME_SOURCEFIELDVALUE</Name>
<System>SYSTEM_NAME</System>
<Search>SEARCH_CRITERIA</Search>
<no_of_sessions>100</no_of_sessions>
</Record_Type>
```

   The following example assumes that the source field is within the Account Business Component.

```
<Parameter>
 <Record_Type>
 <Name>Account_EBIZ</Name>
 <System>SiebelDQ_EBIZ</System>
 <Search>search-org</Search>
 <no_of_sessions>100</no_of_sessions>
 </Record_Type>
</Parameter>

<Parameter>
 <Record_Type>
 <Name>Account_SIEBEL</Name>
 <System>SiebelDQ_SIEBEL</System>
 <Search>search-org</Search>
 <no_of_sessions>100</no_of_sessions>
 </Record_Type>
</Parameter>
```

4. Navigate to the Administration - Data Quality screen, then the Third Party Administration view, and add the new vendor parameter for ISS as shown in the following table (this example assumes that Account is the Business Component Name, and Source is the Field):

| Parameter Name | Value |
|---|---|
| Append Account Record Type Field 1 | Source (this the business component field name where the source information is stored). |

# Sample Configuration Files for Oracle Data Quality Matching Server

This topic provides an example of the ssadq_cfg file which is the configuration file used by Oracle Data Quality Matching Server.

## ssadq_cfg.xml

The ssadq_cfg.xml file is used by Oracle Data Quality Matching Server. An example ssadq_cfg.xml file follows.

```
<?xml version="1.0" encoding="UTF-16"?>
<Data>
 <Parameter>
 <iss_host>hostName</iss_host>
 </Parameter>
 <Parameter>
 <iss_port>1666</iss_port>
 </Parameter>
 <Parameter>
 <rulebase_name>odb:0:userName/passWord@connectString</rulebase_name>
 </Parameter>
 <Parameter>
 <id_tag_name>DQ.RowId</id_tag_name>
 </Parameter>

 <Parameter>
 <Record_Type>
 <Name>Account_Denmark</Name>
 <System>SiebelDQ_Denmark</System>
 <Search>search-org</Search>
 <no_of_sessions>25</no_of_sessions>

 </Record_Type>
 </Parameter>
 <Parameter>
 <Record_Type>
 <Name>Account_USA</Name>
 <System>SiebelDQ_USA</System>
 <Search>search-org</Search>
 <no_of_sessions>25</no_of_sessions>

 </Record_Type>
 </Parameter>
 <Parameter>
 <Record_Type>
 <Name>Account_Germany</Name>
 <System>SiebelDQ_Germany</System>
 <Search>search-org</Search>
 <no_of_sessions>25</no_of_sessions>

 </Record_Type>
 </Parameter>
 <Parameter>
 <Record_Type>
 <Name>Account</Name>
 <System>SiebelDQ</System>
 <Search>search-org</Search>
```

```
<no_of_sessions>25</no_of_sessions>

</Record_Type>
</Parameter>
<Parameter>
<Record_Type>
<Name>Account_China</Name>
<System>siebelDQ_China</System>
<Search>search-org</Search>
<no_of_sessions>25</no_of_sessions>

</Record_Type>
</Parameter>
<Parameter>
<Record_Type>
<Name>Account_Japan</Name>
<System>siebelDQ_Japan</System>
<Search>search-org</Search>
<no_of_sessions>25</no_of_sessions>

</Record_Type>
</Parameter>
<Parameter>
<Record_Type>
<Name>Contact_Denmark</Name>
<System>SiebelDQ_Denmark</System>
<Search>search-person-name</Search>
<no_of_sessions>25</no_of_sessions>

</Record_Type>
</Parameter>
<Parameter>
<Record_Type>
<Name>Contact_USA</Name>
<System>SiebelDQ_USA</System>
<Search>search-person-name</Search>
<no_of_sessions>25</no_of_sessions>

</Record_Type>
</Parameter>
<Parameter>
<Record_Type>
<Name>Contact</Name>
<System>SiebelDQ</System>
<Search>search-person-name</Search>

</Record_Type>
</Parameter>
<Parameter>
<Record_Type>
<Name>Contact_Germany</Name>
<System>SiebelDQ_USA</System>
<Search>search-person-name</Search>
<no_of_sessions>25</no_of_sessions>

</Record_Type>
</Parameter>
<Parameter>
<Record_Type>
<Name>Contact_China</Name>
<System>SiebelDQ_China</System>
<Search>search-person-name</Search>
<no_of_sessions>25</no_of_sessions>

</Record_Type>
</Parameter>
```

```
<Parameter>
<Record_Type>
<Name>Contact_Japan</Name>
<System>SiebelDQ_Japan</System>
<Search>search-person-name</Search>
<no_of_sessions>25</no_of_sessions>

</Record_Type>
</Parameter>
<Parameter>
<Record_Type>
<Name>Prospect</Name>
<System>SiebelDQ</System>
<Search>search-prospect-name</Search>
<no_of_sessions>25</no_of_sessions>

</Record_Type>
</Parameter>
<Parameter>
<Record_Type>
<Name>Prospect_Denmark</Name>
<System>SiebelDQ_Denmark</System>
<Search>search-prospect-name</Search>
<no_of_sessions>25</no_of_sessions>

</Record_Type>
</Parameter>
<Parameter>
<Record_Type>
<Name>Prospect_USA</Name>
<System>SiebelDQ_USA</System>
<Search>search-prospect-name</Search>
<no_of_sessions>25</no_of_sessions>

</Record_Type>
</Parameter>
<Parameter>
<Record_Type>
<Name>Prospect_China</Name>
<System>SiebelDQ_China</System>
<Search>search-prospect-name</Search>
<no_of_sessions>25</no_of_sessions>

</Record_Type>
</Parameter>
<Parameter>
<Record_Type>
<Name>Prospect_Japan</Name>
<System>SiebelDQ_Japan</System>
<Search>search-prospect-name</Search>
<no_of_sessions>25</no_of_sessions>

</Record_Type>
</Parameter>
</Data>
```

# Process of Installing the Oracle Data Quality Address Validation Server

The Oracle Data Quality Address Validation Server provides address validation and standardization functionality using licensed third-party Informatica Identity Resolution software. The process of installing the Oracle Data Quality Address Validation Server for data cleansing is broken down into the following tasks:

1. *Installing Oracle Data Quality Address Validation Server*
2. *Configuring Siebel CRM for the Oracle Data Quality Address Validation Server*
3. *Modifying Configuration Parameters for Oracle Data Quality Address Validation Server*
4. *Acquiring the License Key and Postal Directories for Oracle Data Quality Address Validation Server*

## Installing Oracle Data Quality Address Validation Server

Use the following procedure to install the Oracle Data Quality Address Validation Server. This task is a step in *Process of Installing the Oracle Data Quality Address Validation Server*.

**Note:** Installation is the same no matter what version of Informatica Identity Resolution you are installing.

### To install Oracle Data Quality Address Validation Server

1. Install Informatica Identity Resolution.

   The Oracle Data Quality Address Validation Server is installed as part of Informatica Identity Resolution installation. For more information about Informatica Identity Resolution installation on Microsoft Windows and on UNIX, see the following:

   - *Installing Oracle Data Quality Matching Server*
   - *Configuring Oracle Data Quality Matching Server*

   The ssadqasm.dll file uses the Oracle Data Quality Address Validation Server for address cleansing. You need a license to use the Oracle Data Quality Address Validation Server.
2. Obtain licensing for the postal directories (or postal validation databases), and then:

   **Note:** The postal directories and the license for the postal directories must be obtained directly from Informatica Address Doctor. For more information, see *Acquiring the License Key and Postal Directories for Oracle Data Quality Address Validation Server*. Informatica bundles geographies in different ways - for example, North America is cheaper than USA + Canada + Mexico.

   a. Copy the postal validation databases to the following location:

      `<InstallDir>:/InformaticaIR/ssaas/ad5/ad/db`

      **Note:** This is the postal directory path for Informatica Address Doctor Version 5. For Informatica Address Doctor Version 4, the postal directory path is `InformaticaIR/ssaas/ad/ad/db`.

   b. Copy the following DLLs from `informaticaIR/bin` to `siebelsrvr/bin`:

ssasec.dll

ssadqasm.dll

ssadqsea.dll

ssaiok.dll

> **Note:** Copy these DLLs if using Windows. Copy the libXXXX.so DLLs if using UNIX. For UNIX, the target directory is `siebsrvr/lib`. For Windows, the target directory is `siebsrvr\bin`.]

Make sure that you copy the DLLs to Siebel Server every time you upgrade or apply a new patch for your Siebel application.

c. Place the Oracle Data Quality Address Validation Server key file in the `/ssaas/ad5/ad/db` folder. For example:

```
<InstallDir>:/InformaticaIR/ssaas/ad5/ad/db
```

The `key` file contains an unlock code for specific databases; Informatica sends the `key` file along with the postal directories.

> **Note:** If using Informatica Identity Resolution 9.01, see *Upgrading to Informatica Identity Resolution 9.01* (*Upgrading to Informatica Identity Resolution 9.01*).

3. Install country files in the `/ssaas/ad5/ad/db` folder. For example:

```
/InformaticaIR/ssaas/ad5/ad/db
```

4. Add UNITED STATES to the COUNTRY list of values:

   a. Navigate to Administration - Data screen, then the List of Values Explorer view.
   b. Click Query, and query for the following in the List of Values - Type field: COUNTRY.
   c. In the LOV explorer panel, click the COUNTRY node (by clicking the + sign) and navigate to the values for COUNTRY.
   d. Add a new entry for UNITED STATES.

   Repeat this step for each country where you acquired postal directories. For example, add CANADA to the COUNTRY list of values in the same way, add MEXICO to the COUNTRY list of values in the same way, and so on.

# Modifying Configuration Parameters for Oracle Data Quality Address Validation Server

The ssadq_cfgasm.xml file contains the global configuration parameters for Oracle Data Quality Address Validation Server. You must modify ssadq_cfgasm.xml in order to map Siebel CRM business components (`<bc_field>`) to data types supported by IIR (`<data_type>`). The following table lists the data types that are supported by IIR. For an example ssadq_cfgasm.xml file, see *Sample Configuration Files for Oracle Data Quality Matching Server*.

| Data Type | Meaning |
|---|---|
| Nobility | For example: Lord, Sir, and so on. |
| Title | For example: Mr. Mrs, Dr, and so on. |
| FName | First name |
| MName | Middle name |
| LName | Last name |
| Function | For example: Manager, Director, and so on. |
| Building | Building name |
| SubBuilding | Sub building name |
| HouseNumber | House number |
| Street1 | Street address line 1 |
| Street2 | Street address line 2 |
| POBox | Post office box number |
| DeptLocality | For example: URB, Colonia. |
| Locality | For example: County |
| Province | For example: State |
| Zip | Postal code |
| Country | Country name |

In addition to providing field mappings, the ssadq_cfgasm.xml file defines a standardization operation (`<std_operation>`) for each field, which controls how the field will be standardized.

The following table lists the standardization operations that are supported by IIR.

| Standardization Operation | Description |
|---|---|
| Upper | Convert text to upper case. |

| Standardization Operation | Description |
|---|---|
| | |
| Lower | Convert text to lower case. |
| Camel | Convert text to camel case (upper case for first letter only) |

Use the following procedure to modify configuration parameters for Oracle Data Quality Address Validation Server. This task is a step in *Process of Installing the Oracle Data Quality Address Validation Server*.

## To modify configuration parameters for Oracle Data Quality Address Validation Server

1. Open the ssadq_cfgasm.xml file in a text editor.
2. Use the following syntax to map a Siebel CRM business component field name to a supported IIR data type:

```
<Parameter>
 <datacleanse_mapping>
 <mapping>
 <bc_field>AccountName</bc_field>
 <data_type>Organization</data_type>
 <std_operation>Camel</std_operation>
 </mapping>
 </datacleanse_mapping>
</Parameter>
```

   This example maps the Siebel CRM business component field name *AccountName* to the supported IIR *Organization* data type, and defines *camel* as the standardization operation.

   Repeat this step as required.
3. Save the ssadq_cfgasm.xml file and copy it to the SDQConnector folder on Siebel Server for the changes to take effect:

```
siebsrvr/SDQConnector
```

4. To integrate with the Informatica Address Doctor Version 5 postal directories, add the following tag to the ssadqasm_cfg.xml file located in `siebelserver/SDQConnector`:

```
<Parameter>
 <asm_version>V5</asm_version>
</Parameter>
```

   When enabling data cleansing, you must add the country LOV value according to how the country is returned by the postal directory after cleansing. For example, if Country USA looks like “UNITED STATES” post cleansing, then you must add the LOV value UNITED STATES to the Country picklist.

   > **Note:** An upgrade from Informatica Identity Resolution 2.7.04 to 2.8.07 should be treated like a new setup. In such cases, install Informatica Identity Resolution 2.8.07 on a new port, create a new system, perform the initial load, start the synchronizer to make it operational, and then delete the current Informatica Identity Resolution 2.7.04 setup.

# Acquiring the License Key and Postal Directories for Oracle Data Quality Address Validation Server

The Oracle Data Quality Address Validation Server is installed as part of Informatica Identity Resolution installation. All content and license keys for the postal directories, however, must be purchased directly from Informatica Address Doctor. Subsequent updates and support for the postal directories is provided by Informatica Address Doctor also.

Address Doctor postal directories are currently certified in USA, Canada, and Australia. Address Doctor provides coverage for over 240 countries but not all coverage is the same. Address Doctor assigns a grade (A+, A, B) for each country's coverage. As this grade can change, it is recommended that you check the Address Doctor Web site at the following address for the latest grades:

*http://www.addressdoctor.com/en/countries_data/countries5.asp*

License keys, once purchased, provide a 12-month subscription to the postal directories and restrict the use of address validation to the purchased countries or territories. The maximum duration of the license key is 12 months.

A postal directory is ultimately owned and managed by the country or territory that provides the postal data, and hence is managed differently across providers. You can keep an eye on postal directory updates by:

- Verifying the postal reference data on the Address Doctor Web site.
- Reviewing any update emails that Address Doctor sends.

Use the following procedure to acquire the license key and postal directories for Oracle Data Quality Address Validation Server. This task is a step in *Process of Installing the Oracle Data Quality Address Validation Server*.

## To acquire the license key and postal directories for Oracle Data Quality Address Validation Server

1. Send the following information to Informatica Address Doctor using the email address oracleAV@informatica.com:
    - Full customer contact information, including: company name, contact name, email address, billing address, telephone, and fax numbers.
    - The countries, regions, or territories for which you require the license and postal reference data.
    - The platform on which the Oracle Data Quality Address Validation Server is deployed (for example Windows 32 bit).
    - The underlying Informatica product and version (for example, Informatica Identity Resolution version 2.7 or 2.8).
2. When you have purchased the license key and postal directories for Oracle Data Quality Address Validation Server:
    - Informatica Address Doctor emails the license key information to the named contact.
    - Informatica Address Doctor support emails the credentials to download the reference key to the named contact.

    You need this information to install and access the postal directories as described in *Installing Oracle Data Quality Address Validation Server*.

# Configuring Siebel CRM for the Oracle Data Quality Address Validation Server

Configuring Siebel CRM for the Oracle Data Quality Address Validation Server involves enabling cleansing on all object managers, specifying data cleansing settings, and setting up preconfigured vendor parameters and field mapping values for the Oracle Data Quality Address Validation Server.

Use the following procedure to configure Siebel CRM for the Oracle Data Quality Address Validation Server. This task is a step in *Configuring Oracle Data Quality Matching Server*.

## To configure Siebel CRM for the Oracle Data Quality Address Validation Server

1. Open the uagent.cfg file in a text editor, and modify the [DataCleansing] section of the file to include the following:

   ```
   [DataCleansing]
   Enable=TRUE
   Type=ASM
   ```

   The uagent.cfg file is located in the `Siebel/bin/w32u/enu` directory.
2. In your Siebel application, enable data cleansing to use the Oracle Data Quality Address Validation Server as described in *Enabling and Disabling Data Matching and Data Cleansing*.

   For example, enable data cleansing at the object manager level, enterprise level, user level, and set the data quality settings (for data cleansing). Note that the Data Cleansing Type parameter must be set to ASM as shown in the following table.

| Parameter Name | Value |
|---|---|
| Data Cleansing Enable Flag | True |
| Data Cleansing Type | ASM |

   a. Configure the *ASM vendor applet* (Oracle Data Quality Address Validation Server vendor applet) as shown in the following table by navigating to the Administration - Data Quality screen, then the Third Party Administration view.

| Vendor Applet Parameter Name | Vendor Applet Parameter Value |
|---|---|
| Name | ASM |
| DLL Name | ssadqasm |

| Vendor Applet Parameter Name | Vendor Applet Parameter Value |
|---|---|
| | |

b. Verify that the preconfigured ASM vendor parameter and field mapping values are set up as listed in *Universal Connector Parameter and Field Mapping Values for Oracle Data Quality Address Validation Server*.

c. For better control over the data returned by ASM, add the following vendor parameters for Oracle Data Quality Address Validation Server:

| Vendor Parameter Name | Description |
|---|---|
| ASM Country Database Return Code | Specifies the ASM return codes under this vendor parameter for which any error messages returned are ignored and processing continues if the country database is not found.<br><br>List return codes, separated by a comma. For example, if the customer is using Informatica Address Doctor Version 5 and the country database is not licensed, then specify the vendor parameter as follows:<br><br>`ASM Country Database Return Code: 25,26` |
| ASM High Deliverability Return Code | Specifies the ASM return codes under this vendor parameter for which addresses returned by the ASM Engine override the input address.<br><br>If the ASM return code matches a return code defined within this vendor parameter, then the validated address sent by the ASM Engine is cleansed. In all other cases, the input address is retained.<br><br>List return codes, separated by a comma. For example:<br><br>`ASM High Deliverability Return Code: 0,1,2,3,4,5,6,7,8`<br><br>This vendor parameter applies only if the DQ Cleanse High Deliverable Address vendor parameter is set to Yes. |

3. Modify the ssadq_cfgasm.xml file as described in *Process of Installing the Oracle Data Quality Matching Server*.

# Universal Connector Parameter and Field Mapping Values for Oracle Data Quality Address Validation Server

This topic includes information about the Universal Connector parameters and field mapping values for the Oracle Data Quality Address Validation Server.


- *Preconfigured Vendor Parameters for Oracle Data Quality Address Validation Server*
- *Preconfigured Field Mappings for Oracle Data Quality Address Validation Server*


## Preconfigured Vendor Parameters for Oracle Data Quality Address Validation Server

The following table lists the vendor parameters preconfigured for Oracle Data Quality Address Validation Server. You must not reconfigure the parameter settings.

| Name | Value |
|---|---|
| Account DataCleanse Record Type | Account |
| Contact DataCleanse Record Type | Contact |
| List Mgmt Prospective Contact DataCleanse Record Type | Prospect |
| Personal Address DataCleanse Record Type | Business Address |
| CUT Address DataCleanse Record Type | Business Address |
| DQ Send Empty Field To Third Party Vendor | No |
| DQ Cleanse High Deliverable Address | Yes |
| Parameter 1 | "global", "iss-config-file", "ssadq_cfgasm.xml" |

# Preconfigured Field Mappings for Oracle Data Quality Address Validation Server

This topic includes information about the preconfigured Oracle Data Quality Address Validation Server field mappings for the following business components:

- *Preconfigured Field Mappings for Business Component - Account*
- *Preconfigured Field Mappings for Business Component - Contact*
- *Preconfigured Field Mappings for Business Component - List Mgmt Prospective Contact*
- *Preconfigured Field Mappings for Business Component - CUT Address*
- *Preconfigured Field Mappings for Business Component - Personal Address*

## Preconfigured Field Mappings for Business Component - Account

The following table shows the data cleansing field mappings for the Account business component and data cleansing operation.

| Business Component Field | Mapped Field |
|---|---|
| Name | Account.Name |

## Preconfigured Field Mappings for Business Component - Contact

The following table shows the data cleansing field mappings for the Contact business component and data cleansing operation.

| Business Component Field | Mapped Field |
|---|---|
| First Name | Contact.First Name |
| Last Name | Contact.Last Name |
| Middle Name | Contact.Middle Name |

## Preconfigured Field Mappings for Business Component - List Mgmt Prospective Contact

The following table shows the data cleansing field mappings for the List Mgmt Prospective Contact business component and data cleansing operation.

| Business Component Field | Mapped Field |
|---|---|
| First Name | List Mgmt Prospective Contact.First Name |

| Business Component Field | Mapped Field |
|---|---|
| | |
| Job Title | List Mgmt Prospective Contact.Job Title |
| Last Name | List Mgmt Prospective Contact.Last Name |
| Middle Name | List Mgmt Prospective Contact.Middle Name |

## Preconfigured Field Mappings for Business Component - CUT Address

The following table shows the data cleansing field mappings for the CUT Address business component and data cleansing operation.

**Note:** For Siebel Industry Applications, the CUT Address business component is used instead of the Business Address business component.

| Business Component Field | Mapped Field |
|---|---|
| City | Business Address.City |
| Country | Business Address.Country |
| Postal Code | Business Address.Postal Code |
| State | Business Address.State |
| Street Address | Business Address.Street Address |
| Street Address 2 | Business Address.Street Address 2 |

## Preconfigured Field Mappings for Business Component - Personal Address

The following table shows the data cleansing field mappings for the Personal Address business component and data cleansing operation.

| Business Component Field | Mapped Field |
|---|---|
| City | Business Address.City |
| Country | Business Address.Country |
| Postal Code | Business Address.Postal Code |

| Business Component Field | Mapped Field |
|---|---|
| | |
| State | Business Address.State |
| Street Address | Business Address.Street Address |
| Street Address 2 | Business Address.Street Address 2 |

# Sample Configuration File for Oracle Data Quality Address Validation Server

This topic provides an example of the ssadq_cfgasm.xml file which is the configuration file used by Oracle Data Quality Address Validation Server.

## ssadq_cfgasm.xml

The ssadq_cfgasm.xml file is used by Oracle Data Quality Address Validation Server. An example ssadq_cfgasm.xml file follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<Data>
 <Parameter>
 <iss_host>hostname</iss_host>
 </Parameter>
 <Parameter>
 <iss_port>1666</iss_port>
 </Parameter>
 <Parameter>
 <format_zip>TRUE</format_zip>
 </Parameter>
 <Parameter>
 <datacleanse_mapping>
 <mapping>
 <field>Name</field>
 <ssafield>Organization</ssafield>
 <std_operation>Upper</std_operation>
 </mapping>
 <mapping>
 <field>Street_spcAddress</field>
 <ssafield>Street1</ssafield>
 <std_operation>Upper</std_operation>
 </mapping>
 <mapping>
 <field>City</field>
 <ssafield>Locality</ssafield>
 </mapping>
 <mapping>
 <field>Postal_spcCode</field>
 <ssafield>Zip</ssafield>
 </mapping>
 <mapping>
```

```
<field>State</field>
<ssafield>Province</ssafield>
</mapping>
<mapping>
<field>Country</field>
<ssafield>Country</ssafield>
</mapping>
<mapping>
<field>First_spcName</field>
<ssafield>FName</ssafield>
<std_operation>Upper</std_operation>
</mapping>
<mapping>
<field>Middle_spcName</field>
<ssafield>MName</ssafield>
<std_operation>Upper</std_operation>
</mapping>
<mapping>
<field>Last_spcName</field>
<ssafield>LName</ssafield>
<std_operation>Upper</std_operation>
</mapping>
<mapping>
<field>Personal_spcPostal_spcCode</field>
<ssafield>Zip</ssafield>
</mapping>
<mapping>
<field>Personal_spcCity</field>
<ssafield>Locality</ssafield>
</mapping>
<mapping>
<field>Personal_spcState</field>
<ssafield>Province</ssafield>
</mapping>
<mapping>
<field>Personal_spcStreet_spcAddress</field>
<ssafield>Street1</ssafield>
<std_operation>Camel</std_operation>
</mapping>
<mapping>
<field>Personal_spcStreet_spcAddress 2</field>
<ssafield>Street2</ssafield>
<std_operation>Camel</std_operation>
</mapping>
<mapping>
<field>Personal_spcCountry</field>
<ssafield>Country</ssafield>
</mapping>
</datacleanse_mapping>
</Parameter>
</Data>
```

# Siebel CRM ISSSYNC Action Sets

ISSSYNC action sets use Siebel application workflows to send messages to the Oracle Data Quality Matching Server XS Server (XML Sync Server). Activate ISSSYNC action sets if using a code base earlier than Siebel CRM Version 8.2. For more information about creating action sets, including creating actions for action sets, and associating events with action sets, see *Siebel Personalization Administration Guide* .

This topic introduces the following ISSSYNC action sets for Siebel CRM:

- *ISSSYNC Action Sets for Account*
- *ISSSYNC Action Sets for Contact*
- *ISSSYNC Action Sets for List Mgmt Prospective Contact*
- *Generic ISSSYNC Action Sets*

# ISSSYNC Action Sets for Account

This topic introduces the following Siebel application ISSSYNC action sets for Account:

- *ISSLoad Account*
- *ISSSYNC DeleteRecord Account*
- *ISSSYNC PreDeleteRecord Account*
- *ISSSYNC PreWriteRecord Account*
- *ISSSYNC WriteRecord Account*

## ISSLoad Account

The following table describes the actions in the ISSLoad Account action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | SiebelDQ |
| ISS Set Page Size | Name | ISS Set Page Size |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_PAGE_SIZE |
| | Set Operator | Set |

| Action | Name of Field | Value |
|---|---|---|
| | Value | 80 |
| ISS Set File Name | Name | ISS Set File Name |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_LOADFILE |
| | Set Operator | Set |
| | Value | "C:\ids\iss2704s\ids\data\account.xml"<br><br>**Note:** Modify this value if you install Oracle Data Quality Matching Server on a drive other than `C:\` drive. |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_ACCOUNT |
| Set IO Name | Name | ISS Set IO Name |
| | Sequence | 5 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_NAME |
| | Set Operator | Set |
| | Value | ISS_Account |
| ISS Run WF | Name | ISS Run WF |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Sequence | 6 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS Launch Build Load File" |

## ISSSYNC DeleteRecord Account

The following table describes the actions in the ISSSYNC DeleteRecord Account action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set URL | Name | ISS Set URL |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_URL |
| | Set Operator | Set |
| | Value | "http://SERVERNAME:1671"<br><br>**Note:** Replace SERVERNAME with the Hostname or IP address of the computer where XML Sync Server (XS) is installed. |
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 2 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |

| Action | Name of Field | Value |
|---|---|---|
| | Business Service Context | "ProcessName", "ISS Launch Delete Record Sync" |

## ISSSYNC PreDeleteRecord Account

The following table describes the actions in the ISSSYNC PreDeleteRecord Account action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | SiebelDQ |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_ACCOUNT |
| ISS Set IO Name | Name | ISS Set IO Name |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_NAME |
| | Set Operator | Set |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Value | ISS_Account |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS Launch PreDelete Record Sync" |

## ISSSYNC PreWriteRecord Account

The following table describes the actions in the ISSSYNC PreWriteRecord Account action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |

| Action | Name of Field | Value |
|---|---|---|
| | Set Operator | Set |
| | Value | SiebelDQ |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_ACCOUNT |
| ISS Set IO Name | Name | ISS Set IO Name |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_NAME |
| | Set Operator | Set |
| | Value | ISS_Account |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Run WF | Name | ISS Run WF |

| Action | Name of Field | Value |
|---|---|---|
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS Launch PreWrite Record Sync" |

## ISSSYNC WriteRecord Account

The following table describes the actions in the ISSYNC WriteRecord Account action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | SiebelDQ |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_ACCOUNT |
| ISS Set IO Name | Name | ISS Set IO Name |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_NAME |
| | Set Operator | Set |
| | Value | ISS_Account |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Set URL | Name | ISS Set URL |
| | Sequence | 5 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_URL |
| | Set Operator | Set |
| | Value | "http://SERVERNAME:1671"<br><br>**Note:** Replace SERVERNAME with the Hostname or IP address of the computer where XML Sync Server (XS) is installed. |
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 6 |

| Action | Name of Field | Value |
|---|---|---|
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS Launch Write Record Sync" |

# ISSSYNC Action Sets for Contact

This topic introduces the following Siebel application ISSSYNC action sets for Contact:

- *ISSLoad Contact*
- *ISSSYNC DeleteRecord Contact*
- *ISSSYNC PreDeleteRecord Contact*
- *ISSSYNC PreWriteRecord Contact*
- *ISSSYNC WriteRecord Contact*

## ISSLoad Contact

The following table describes the actions in the ISSLoad Contact action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | SiebelDQ |
| ISS Set Page Size | Name | ISS Set Page Size |
| | Sequence | 2 |
| | Action Type | Attribute Set |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Profile Attribute | IDS_PAGE_SIZE |
| | Set Operator | Set |
| | Value | 80 |
| ISS Set File Name | Name | ISS Set File Name |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_LOADFILE |
| | Set Operator | Set |
| | Value | "C:\ids\iss2704s\ids\data\contact.xml"<br><br>**Note:** Modify this value if you install Oracle Data Quality Matching Server on a drive other than `C:\` drive. |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_CONTACT |
| ISS Set IO Name | Name | ISS Set IO Name |
| | Sequence | 5 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_NAME |

| Action | Name of Field | Value |
|---|---|---|
| | Set Operator | Set |
| | Value | ISS_Contact |
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 6 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS Launch Build Load File" |

## ISSSYNC DeleteRecord Contact

The following table describes the actions in the ISSSYNC DeleteRecord Contact action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set URL | Name | ISS Set URL |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_URL |
| | Set Operator | Set |
| | Value | "http://SERVERNAME:1671"<br><br>**Note:** Replace SERVERNAME with the Hostname or IP address of the computer where XML Sync Server (XS) is installed. |
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 2 |

| Action | Name of Field | Value |
|---|---|---|
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS Launch Delete Record Sync" |

## ISSSYNC PreDeleteRecord Contact

The following table describes the actions in the ISSYNC PreDeleteRecord Contact action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | SiebelDQ |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_CONTACT |
| ISS Set IO Name | Name | ISS Set IO Name |
| | Sequence | 3 |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_NAME |
| | Set Operator | Set |
| | Value | ISS_Contact |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS Launch PreDelete Record Sync" |

## ISSSYNC PreWriteRecord Contact

The following table describes the actions in the ISSSYNC PreWriteRecord Contact action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |

| Action | Name of Field | Value |
|---|---|---|
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | SiebelDQ |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_CONTACT |
| ISS Set IO Name | Name | ISS Set IO Name |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_NAME |
| | Set Operator | Set |
| | Value | ISS_Contact |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |

| Action | Name of Field | Value |
|---|---|---|
| | Set Operator | Set |
| | Value | [Id] |
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS Launch PreWrite Record Sync" |

## ISSSYNC WriteRecord Contact

The following table describes the actions in the ISSSYNC WriteRecord Contact action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | SiebelDQ |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Set Operator | Set |
| | Value | IDS_01_IDT_CONTACT |
| ISS Set IO Name | Name | ISS Set IO Name |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_NAME |
| | Set Operator | Set |
| | Value | ISS_Contact |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Set URL | Name | ISS Set URL |
| | Sequence | 5 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_URL |
| | Set Operator | Set |
| | Value | "http://SERVERNAME:1671" |

| Action | Name of Field | Value |
|---|---|---|
| | | **Note:** Replace SERVERNAME with the Hostname or IP address of the computer where XML Sync Server (XS) is installed. |
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 6 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS Launch Write Record Sync" |

# ISSSYNC Action Sets for List Mgmt Prospective Contact

This topic introduces the following Siebel application ISSSYNC action sets for List Mgmt Prospective Contact:

- *ISSLoad Prospect*
- *ISSSYNC DeleteRecord Prospect*
- *ISSSYNC PreDeleteRecord Prospect*
- *ISSSYNC PreWriteRecord Prospect*
- *ISSSYNC WriteRecord Prospect*

## ISSLoad Prospect

The following table describes the actions in the ISSLoad Prospect action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |

| Action | Name of Field | Value |
|---|---|---|
| | Value | SiebelDQ |
| ISS Set Page Size | Name | ISS Set Page Size |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_PAGE_SIZE |
| | Set Operator | Set |
| | Value | 80 |
| ISS Set File Name | Name | ISS Set File Name |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_LOADFILE |
| | Set Operator | Set |
| | Value | "C:\ids\iss2704s\ids\data\prospect.xml"<br><br>**Note:** Modify this value if you install Oracle Data Quality Matching Server on a drive other than `C:\` drive. |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_PROSPECT |
| ISS Set IO Name | Name | ISS Set IO Name |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Sequence | 5 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_NAME |
| | Set Operator | Set |
| | Value | ISS_List_Mgmt_Prospective_Contact |
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 6 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS Launch Build Load File" |

## ISSSYNC DeleteRecord Prospect

The following table describes the actions in the ISSSYNC DeleteRecord Prospect action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set URL | Name | ISS Set URL |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_URL |
| | Set Operator | Set |
| | Value | "http://SERVERNAME:1671" |

| Action | Name of Field | Value |
|---|---|---|
| | | **Note:** Replace SERVERNAME with the Hostname or IP address of the computer where XML Sync Server (XS) is installed. |
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 2 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS Launch Delete Record Sync" |

## ISSSYNC PreDeleteRecord Prospect

The following table describes the actions in the ISSSYNC PreDeleteRecord Prospect action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | SiebelDQ |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Value | IDS_01_IDT_PROSPECT |
| ISS Set IO Name | Name | ISS Set IO Name |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_NAME |
| | Set Operator | Set |
| | Value | ISS_List_Mgmt_Prospective_Contact |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS Launch PreDelete Record Sync" |

## ISSSYNC PreWriteRecord Prospect

The following table describes the actions in the ISSSYNC PreWriteRecord Prospect action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | SiebelDQ |
| ISS Set IDT Name | Name | ISS Set IDT Name |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_PROSPECT |
| ISS Set IO Name | Name | ISS Set IO Name |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_NAME |
| | Set Operator | Set |
| | Value | ISS_List_Mgmt_Prospective_Contact |
| ISS Set ID | Name | ISS Set ID |

| Action | Name of Field | Value |
|---|---|---|
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 5 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS Launch PreWrite Record Sync" |

## ISSSYNC WriteRecord Prospect

The following table describes the actions in the ISSSYNC WriteRecord Prospect action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Set System Name | Name | ISS Set System Name |
| | Sequence | 1 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_SYSTEM |
| | Set Operator | Set |
| | Value | SiebelDQ |
| ISS Set IDT Name | Name | ISS Set IDT Name |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Sequence | 2 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IDT |
| | Set Operator | Set |
| | Value | IDS_01_IDT_PROSPECT |
| ISS Set IO Name | Name | ISS Set IO Name |
| | Sequence | 3 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_NAME |
| | Set Operator | Set |
| | Value | ISS_List_Mgmt_Prospective_Contact |
| ISS Set ID | Name | ISS Set ID |
| | Sequence | 4 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_IO_ID |
| | Set Operator | Set |
| | Value | [Id] |
| ISS Set URL | Name | ISS Set URL |
| | Sequence | 5 |
| | Action Type | Attribute Set |
| | Profile Attribute | IDS_URL |

| Action | Name of Field | Value |
|---|---|---|
| | | |
| | Set Operator | Set |
| | Value | "http://SERVERNAME:1671"<br><br>**Note:** Replace SERVERNAME with the Hostname or IP address of the computer where XML Sync Server (XS) is installed. |
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 6 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| ISS Run WF (continued) | Business Service Context | "ProcessName", "ISS Launch Write Record Sync" |

# Generic ISSSYNC Action Sets

This topic introduces the Siebel application generic ISSSYNC action sets for all business components:

- *ISSSYNC WriteRecordNew*
- *ISSSYNC WriteRecordUpdated*

## ISSSYNC WriteRecordNew

The following table describes the actions in the ISSSYNC WriteRecordNew action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 1 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |

| Action | Name of Field | Value |
|---|---|---|
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS WriteRecordNew" |

### ISSSYNC WriteRecordUpdated

The following table describes the actions is in the ISSSYNC WriteRecordUpdated action set.

| Action | Name of Field | Value |
|---|---|---|
| ISS Run WF | Name | ISS Run WF |
| | Sequence | 1 |
| | Action Type | BusService |
| | Business Service Name | Workflow Process Manager |
| | Business Service Method | RunProcess |
| | Business Service Context | "ProcessName", "ISS WriteRecordUpdated" |

## Activating Action Sets in Your Siebel Application

Use the following procedure to activate action sets in your Siebel application.

### To activate action sets in your Siebel application

1. Verify that the action sets for Account, Contact, and List Mgmt Prospective Contact are set up in your Siebel application by navigating to Administration - Runtime Events, then the Action Sets view.

   For information about the action sets that are set up by default in your Siebel application, see *Siebel CRM ISSSYNC Action Sets*. For more information about creating action sets, including creating actions for action sets, see *Siebel Personalization Administration Guide* .

   > **Note:** When verifying ISSSYNC action set setup, make sure that the IDS_URL profile attribute reflects the URL location of Oracle Data Quality Matching Server.

2. Verify that appropriate run-time events (seed data) are set up in your Siebel application by navigating to Administration - Runtime Events, then the Events view.
   For more information about run-time events, including how to call a workflow process from a run-time event, see *Siebel Business Process Framework: Workflow Guide* .
   For more information about associating events with action sets, see *Siebel Personalization Administration Guide* .
3. Activate the action sets for Account, Contact, and List Mgmt Prospective Contact, as follows:
   a. Navigate to Administration - Runtime Events, then the Action Sets view.
   b. Select the Active checkbox for each Action Set that you want to activate.
   c. Reload the run-time events by clicking Menu, and selecting Reload Runtime Events.

# Sample SQL Scripts

This topic provides examples of the SQL scripts that are used for incremental data load.

- *IDS_IDT_ACCOUNT_STG.SQL*
- *IDS_IDT_CONTACT_STG.SQL*
- *IDS_IDT_PROSPECT_STG.SQL*
- *IDS_IDT_CURRENT_BATCH.SQL*
- *IDS_IDT_CURRENT_BATCH_ACCOUNT.SQL*
- *IDS_IDT_CURRENT_BATCH_CONTACT.SQL*
- *IDS_IDT_CURRENT_BATCH_PROSPECT.SQL*
- *IDS_IDT_LOAD_ANY_ENTITY.CMD* (Windows)
- *IDS_IDT_LOAD_ANY_ENTITY.sh* (UNIX)

## IDS_IDT_ACCOUNT_STG.SQL

The following sample SQL script can be used for incremental data load.

```
/*
'============================================================================='
' Need to change TBLO before executing the scripts on target database. '
'============================================================================='
*/
SET TERMOUT ON
SET FEEDBACK OFF
SET VERIFY OFF
SET TIME OFF
SET TIMING OFF
SET ECHO OFF
SET PAUSE OFF
DROP MATERIALIZED VIEW ACCOUNTS_SNAPSHOT_VIEW;
CREATE MATERIALIZED VIEW ACCOUNTS_SNAPSHOT_VIEW AS
SELECT
 T2.ROW_ID ACCOUNT_ID,
 T2.NAME ACCOUNT_NAME,
 T3.ROW_ID ACCOUNT_ADDR_ID,
 T3.ADDR ADDRESS_LINE1,
 T3.ADDR_LINE_2 ADDRESS_LINE2,
```

```
 T3.COUNTRY COUNTRY,
 T3.STATE STATE,
 T3.CITY CITY,
 T3.ZIPCODE POSTAL_CODE,
 DECODE(T2.PR_BL_ADDR_ID,T3.ROW_ID,'Y','N') PRIMARY_FLAG,
 FLOOR((ROWNUM-1)/&BATCH_SIZE)+1 BATCH_NUM
 FROM
 dbo.S_CON_ADDR T1,
 dbo.S_ORG_EXT T2,
 dbo.S_ADDR_PER T3
 WHERE
 T1.ACCNT_ID = T2.ROW_ID
 AND
 T1.ADDR_PER_ID = T3.ROW_ID
-- Comment the following line for Multiple address match option
-- AND T2.PR_BL_ADDR_ID=T3.ROW_ID
/
SELECT '================================================================================='
|| CHR(10) ||
 ' REPORT ON ACCOUNTS SNAPSHOT' || CHR(10) ||
 '================================================================================='
|| CHR(10) " "
 FROM DUAL
/
SELECT BATCH_NUM BATCH, COUNT(*) "NUMBER OF RECORDS"
FROM ACCOUNTS_SNAPSHOT_VIEW
GROUP BY BATCH_NUM
ORDER BY BATCH_NUM
/
```

# IDS_IDT_CONTACT_STG.SQL

The following sample SQL script can be used for incremental data load.

```
/*
================================================================================
 Need to change TBLO before executing the scripts on target database.
================================================================================
*/
SET TERMOUT ON
SET FEEDBACK OFF
SET VERIFY OFF
SET TIME OFF
SET TIMING OFF
SET ECHO OFF
SET PAUSE OFF
SET PAGESIZE 50
DROP MATERIALIZED VIEW CONTACTS_SNAPSHOT_VIEW;
CREATE MATERIALIZED VIEW CONTACTS_SNAPSHOT_VIEW AS
SELECT
 T1.CONTACT_ID CONTACT_ID,
 T2.FST_NAME || ' ' || LAST_NAME NAME,
 T2.MID_NAME MIDDLE_NAME,
 T3.ROW_ID ADDRESS_ID,
 T3.CITY CITY,
 T3.COUNTRY COUNTRY,
 T3.ZIPCODE POSTAL_CODE,
 T3.STATE STATE,
 T3.ADDR STREETADDRESS,
 T3.ADDR_LINE_2 ADDRESS_LINE2,
 DECODE(T2.PR_PER_ADDR_ID,T3.ROW_ID,'Y','N') PRIMARY_FLAG,
 T4.NAME ACCOUNT,
```

```
 T2.BIRTH_DT BirthDate,
 T2.CELL_PH_NUM CellularPhone,
 T2.EMAIL_ADDR EmailAddress,
 T2.HOME_PH_NUM HomePhone,
 T2.SOC_SECURITY_NUM SocialSecurityNumber,
 T2.WORK_PH_NUM WorkPhone,
 FLOOR((ROWNUM-1)/&BATCHSIZE)+1 BATCH_NUM
FROM
 dbo.S_CON_ADDR T1,
 dbo.S_CONTACT T2,
 dbo.S_ADDR_PER T3,
 dbo.S_ORG_EXT T4
WHERE
 T1.CONTACT_ID = T2.ROW_ID AND
 T1.ADDR_PER_ID = T3.ROW_ID AND
-- OR (T1.ADDR_PER_ID IS NULL)) Do we need contacts with no address?
--Comment the following line for Multiple address match option
-- T2.PR_PER_ADDR_ID = T3.ROW_ID (+) AND
 T2.PR_DEPT_OU_ID = T4.PAR_ROW_ID (+)
/
SELECT '================================================================================'
|| CHR(10) ||
 ' REPORT ON CONTACTS SNAPSHOT' || CHR(10) ||
 '================================================================================'
|| CHR(10) " "
 FROM DUAL
/
SELECT BATCH_NUM BATCH, COUNT(*) "NUMBER OF RECORDS"
FROM CONTACTS_SNAPSHOT_VIEW
GROUP BY BATCH_NUM
ORDER BY BATCH_NUM
/
```

# IDS_IDT_PROSPECT_STG.SQL

The following sample SQL script can be used for incremental data load.

```
/*
'================================================================================'
' Need to change TBLO before executing the scripts on target database. '
'================================================================================'
*/
SET TERMOUT ON
SET FEEDBACK OFF
SET VERIFY OFF
SET TIME OFF
SET TIMING OFF
SET ECHO OFF
SET PAUSE OFF
SET PAGESIZE 50
DROP MATERIALIZED VIEW PROSPECTS_SNAPSHOT_VIEW;
CREATE MATERIALIZED VIEW PROSPECTS_SNAPSHOT_VIEW AS
SELECT
 CON_PR_ACCT_NAME ACCOUNT_NAME,
 CELL_PH_NUM CELLULAR_PHONE,
 CITY CITY,
 COUNTRY COUNTRY,
 EMAIL_ADDR EMAIL_ADDRESS,
 FST_NAME || ' ' || LAST_NAME NAME,
 HOME_PH_NUM HOME_PHONE,
 MID_NAME MIDDLE_NAME,
 ZIPCODE POSTAL_CODE,
```

```
 SOC_SECURITY_NUM SOCIAL_SECURITY_NUMBER,
 STATE STATE,
 ADDR STREETADDRESS,
 ADDR_LINE_2 ADDRESS_LINE2,
 WORK_PH_NUM WORK_PHONE,
 ROW_ID PROSPECT_ID,
 FLOOR((ROWNUM-1)/&BATCH_SIZE)+1 BATCH_NUM
FROM
 dbo.S_PRSP_CONTACT T2
/
SELECT '================================================================================'
|| CHR(10) ||
 ' REPORT ON PROSPECTS SNAPSHOT' || CHR(10) ||
 '================================================================================'
|| CHR(10) " "
 FROM DUAL
/
SELECT BATCH_NUM BATCH, COUNT(*) "NUMBER OF RECORDS"
FROM PROSPECTS_SNAPSHOT_VIEW
GROUP BY BATCH_NUM
ORDER BY BATCH_NUM
/
```

# IDS_IDT_CURRENT_BATCH.SQL

The following sample SQL script can be used for incremental data load.

```
SET FEEDBACK ON
DROP TABLE IDS_IDT_CURRENT_BATCH
/
CREATE TABLE IDS_IDT_CURRENT_BATCH
 ( BATCH_NUM INTEGER)
/
INSERT INTO IDS_IDT_CURRENT_BATCH VALUES (1)
/
```

# IDS_IDT_CURRENT_BATCH_ACCOUNT.SQL

The following sample SQL script can be used for incremental data load.

```
CREATE OR REPLACE VIEW INIT_LOAD_ALL_ACCOUNTS AS
SELECT
 ACCOUNT_ID,
 ACCOUNT_NAME,
 ACCOUNT_ADDR_ID,
 ADDRESS_LINE1,
 ADDRESS_LINE2,
 COUNTRY,
 STATE,
 CITY,
 POSTAL_CODE,
 PRIMARY_FLAG
FROM
 ACCOUNTS_SNAPSHOT_VIEW
WHERE
 BATCH_NUM=
(SELECT BATCH_NUM FROM IDS_IDT_CURRENT_BATCH)
/
```

# IDS_IDT_CURRENT_BATCH_CONTACT.SQL

The following sample SQL script can be used for incremental data load.

```
CREATE OR REPLACE VIEW INIT_LOAD_ALL_CONTACTS AS
SELECT
 CONTACT_ID,
 NAME,
 MIDDLE_NAME,
 ADDRESS_ID,
 CITY,
 COUNTRY,
 POSTAL_CODE,
 STATE,
 STREETADDRESS,
 ADDRESS_LINE2,
 PRIMARY_FLAG,
 BirthDate,
 CellularPhone,
 EmailAddress,
 HomePhone,
 SocialSecurityNumber,
 WorkPhone,
 ACCOUNT
FROM
 CONTACTS_SNAPSHOT_VIEW
WHERE
 BATCH_NUM=
(SELECT BATCH_NUM FROM IDS_IDT_CURRENT_BATCH)
/
```

# IDS_IDT_CURRENT_BATCH_PROSPECT.SQL

The following sample SQL script can be used for incremental data load.

```
CREATE OR REPLACE VIEW INIT_LOAD_ALL_PROSPECTS AS
SELECT
 ACCOUNT_NAME,
CELLULAR_PHONE,
 CITY,
 COUNTRY,
 EMAIL_ADDRESS,
 NAME,
 HOME_PHONE,
 MIDDLE_NAME,
 POSTAL_CODE,
 SOCIAL_SECURITY_NUMBER,
 STATE,
 STREETADDRESS,
 WORK_PHONE,
 PROSPECT_ID
FROM
 PROSPECTS_SNAPSHOT_VIEW
WHERE
 BATCH_NUM=
(SELECT BATCH_NUM FROM IDS_IDT_CURRENT_BATCH)
/
```

# IDS_IDT_LOAD_ANY_ENTITY.CMD

The following sample SQL script can be used for incremental data load.

**Note:** Use this file for Microsoft Windows.

```
@echo off
REM ************************************************************************
REM * *
REM * 1. Change informaticaHome to point to your IIR installation folder *
REM * 2. Change initLoadScripts to point to your Initial Load scripts *
REM * *
REM ************************************************************************
 if %1.==. goto Error
 if %2.==. goto Error
 if %3.==. goto Error
 NOT %4.==. goto GIvenBatchOnly
REM ************************************************************************
REM * *
REM * Setting parameters *
REM * *
REM ************************************************************************
set current=%1
set workdir=%2
set dbcredentials=%3
set machineName=%computername%
set informaticaHome=C:\InformaticaIR
set initLoadScripts=C:\InformaticaIR\InitLoadScripts
REM ************************************************************************
REM * *
REM * Find the number of batches in the current Entity records snapshot *
REM * *
REM ************************************************************************
FOR /F "usebackq delims=!" %%i IN (`sqlplus -s %dbcredentials% @GetBatchCount%1`) DO set
xresult=%%i
set /a NumBatches=%xresult%
echo %NumBatches%
del /s/f/q %workdir%\*
setlocal enabledelayedexpansion
set /a counter=1
REM ************************************************************************
REM * *
REM * Loop through all the batches *
REM * *
REM ************************************************************************
for /l %%a in (2, 1, !NumBatches!) do (
 set /a counter += 1
 (echo counter=!counter!)
 sqlplus %dbcredentials% @%initLoadScripts%\SetBatchNumber.sql !counter!
 cd /d %informaticaHome%\bin
 idsbatch -h%machineName%:1669 -i%initLoadScripts%\idt_%current%_load.txt -
1%workdir%\idt_%current%_load!counter!.log -2%workdir%\idt_%current%_load!counter!.err
-3%workdir%\idt_%current%_load!counter!.dbg
)
goto DONE
:GivenBatchOnly
 echo Processing Batch %4....
 sqlplus %dbcredentials% @%initLoadScripts%\SetBatchNumber.sql %4
 cd /d %informaticaHome%\bin
 idsbatch -h%machineName%:1669 -i%initLoadScripts%\idt_%current%_load.txt -
```

```
1%workdir%\idt_%current%_load%4.log -2%workdir%\idt_%current%_load%4.err -
3%workdir%\idt_%current%_load%4.dbg
 goto DONE
:Error
 ECHO Insufficient parameters
 echo usage "IDS_IDT_LOAD_ANY_ENTITY.CMD <Object_Name> <Work_Dir> <DBUser/
DBPassword@TNS_Entry_Name> [Optional Batch Number]"
 ECHO
 echo e.g. IDS_IDT_LOAD_ANY_ENTITY.CMD ACCOUNT C:\InformaticaIR\InitLoadScripts
ora1234/ora1234@ora_db
 GOTO END
:DONE
 Echo Process completed. Please examine error and log files in %workdir%
 OFF
 set /a errorcount = 0
 %%R in (*.err) do if %%~zR neq 0 set /a errorcount += 1
 error Count =%errorcount%
 if %errorcount% neq 0 goto batcherror
 GOTO END
:batcherror
 echo %errorcount% batch/es have failed. Please check the following batches:
 for %%R in (*.err) do if %%~zR neq 0 echo %%R
 goto DONE
:END
```

# IDS_IDT_LOAD_ANY_ENTITY.sh

The following sample SQL script can be used for incremental data load.

**Note:** Use this file for UNIX.

```
#!/bin/bash
################################################################################
# Prerequisite check block #
################################################################################
# Checking IIR system variables are set. If not then throw error and exit.
if [ -z "$SSABIN" ] && [ -z "$SSATOP" ]
then
 echo "Err #LOAD-01: Informatica IIR system variables not set. Please use 'idsset'
script"
 exit
else
 # checking if required idsbatch utility exists at $SSABIN location
 if [ -f $SSABIN/idsbatch ]
 then
 echo "idsbatch utility found."
 fi
fi
################################################################################
# Param block #
################################################################################
# INPUT PARAMETERS
 current=$1
 workdir=$2
 dbcredentials=$3
# ENVIRONMENT RELATED PARAMETERS
# scriptdir=/export/home/qa1/InformaticaIR/initloadscripts
# informaticadir=/export/home/qa1/InformaticaIR
 scriptdir=$SSATOP/initloadscripts
 =$SSATOP
```

```
# DEBUG OPTION - 1 for ON and 0 for OFF
 debug=1

# Passing DB credentials as argument
# ISS DATABASE CREDENTIALS and CONNECT INFO
# dbcredentials=ora32155/ora32155@sdchs20n532_qa532a
 dbcredentials=$3
# MACHINE NAME
 machineName=`hostname`
################################################################################
# Execution block #
################################################################################
if [ $debug -eq 1 ]; then
 echo using Script dir: $scriptdir
 using Informatica Home: $informaticadir
fi
if [ $# -lt 3 ]
then
 echo "Err #LOAD-03: Error in $0 - Invalid Argument Count"
 echo Usage LoadAnyEntity "<Entity Account,Contact or Prospect> <WorkDir> <dbuser/
dbpass@tnsname> [Optional Batch Number]"
 echo Insufficient parameters
 echo e.g "Syntax: $0 Entity_name Account Log_directory /temp"
 exit
fi
if [ -f $scriptdir/idt_$current\_load.txt ]
 then
 if [ $debug -eq 1 ]; then
 echo Using Load file $scriptdir/idt_$current\_load.txt
 fi
 else
 Load file cannot be loaded. Please check and rerun process
fi
if [ $# -eq 4 ]
 then
 Specific bath to be loaded: $4
fi
if [ -d $workdir ]; then
 cd $workdir
 rm -r -f *.err
fi
if [ $# -eq 3 ]
 then
 read_sql_stmt() {
 typeset stmt=$1
 typeset login=$2

 echo "
 set feedback off verify off heading off pagesize 0
 $stmt;
 exit
 " | sqlplus -s $login
 }

 read_sql_stmt "select max(batch_num) from "$current"s_SNAPSHOT_VIEW"
"$dbcredentials" | while read u
 do

 batches=$u

 counter=2
 if [ $debug -eq 1 ]; then
 echo current=$current
 echo workdir=$workdir
 echo counter=$counter
 echo number of batches to be processed is: $batches
```

```
 fi

# for counter in $(seq 2 $batches);
 for ((counter=2; counter <= $batches; counter++));

currentbatch=$(
sqlplus -S $dbcredentials <<!
set head off feedback off echo off pages 0
UPDATE IDS_IDT_CURRENT_BATCH set batch_num=$counter
/
select batch_num from IDS_IDT_CURRENT_BATCH
/
!
 echo
 echo
 echo "########################################"
 echo "# Curently Processing Batch: $currentbatch #"
 echo "########################################"
 cd $informaticadir/bin
 if [ $debug -eq 1 ]; then
 echo InformaticaDrive: ${PWD}
 echo Processing following command:
 echo idsbatch -h$machineName:1669 -i$scriptdir/idt_$current\_load.txt -1$workdir/
idt_$current\_load$counter.log -2$workdir/idt_$current\_load$counter.err -
3$workdir\idt_$current\_load$counter.dbg
 echo "########################################"
 fi
 idsbatch -h$machineName:1669 -i$scriptdir/idt_$current\_load.txt -1$workdir/
idt_$current\_load$counter.log -2$workdir/idt_$current\_load$counter.err -
3$workdir\idt_$current\_load$counter.dbg
 done
 done
else
 counter=$4
 echo "########################################"
 echo Processing Batch $4....
 currentbatch=$(
 sqlplus -S $dbcredentials <<!
 set head off feedback off echo off pages 0
 UPDATE IDS_IDT_CURRENT_BATCH set batch_num=$counter
 /
 select batch_num from IDS_IDT_CURRENT_BATCH
 /
 !
 )
 echo "########################################"
 cd $informaticadrive/bin
 idsbatch -h$machineName:1669 -i$scriptdir/idt_$current\_load.txt -1$workdir/
idt_$current\_load$counter.log -2$workdir/idt_$current\_load$counter.err -
3$workdir\idt_$current\_load$counter.dbg
fi
 echo "Process completed. Please examine error and log files in "$workdir
 # errorcnt=0
if [ -d $workdir ]; then
 cd $workdir
fi
 errorcnt=$(find ./ -depth 1 -name "*.err" ! -size 0 | wc -l)
echo Errors encountered is: $errorcnt
if [ $errorcnt -eq 0 ]; then
 echo Successfully processed all the batches
 else
 echo ########################################
 echo # Failed batch report #
 echo ########################################
 echo $errorcnt batch/es have failed. Please check the following batches:
 find ./ -depth 1 -name "*.err"
```

```
fi
```

# Sample SiebelDQ.sdf File

The following is an example of a SiebelDQ.sdf file.

```
Section: System
*-------------------------------------------------------------------------
*
*************
* Create a System for each Country. Use separate SDF files for each Country. Use Smallar
case for System Name.
*************
system-definition
*==================
NAME= siebeldq
ID= s1
DEFAULT-PATH= "+"
*
idt-definition
*==============
NAME= IDT_ACCOUNT
*
idt-definition
*==============
NAME= IDT_CONTACT
*
idt-definition
*==============
NAME= IDT_PROSPECT
*
idx-definition
*==============
NAME= IDX_ACCOUNT
ID= 1s
IDT-NAME= IDT_ACCOUNT
KEY-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("FIELD=Organization_Name
 KEY_LEVEL=Standard"),
 Field(Name), Null-Key("K$$$$$$$")
OPTIONS= No-Null-Key,
 Compress-Key-Data(150)
*
idx-definition
*==============
NAME= IDX_CONTACT_NAME
ID= 2s
IDT-NAME= IDT_CONTACT
KEY-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("FIELD=Person_Name KEY_LEVEL=Standard"),
 Field(Name),
 Null-Key("K$$$$$$$")
OPTIONS= No-Null-Key,
 Compress-Key-Data(150)
*
idx-definition
*==============
```

```
NAME= IDX_CONTACT_ADDR
ID= 3s
IDT-NAME= IDT_CONTACT
KEY-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("FIELD=Address_part1 KEY_LEVEL=Standard"),
 Field(StreetAddress),
 Null-Key("K$$$$$$$")
OPTIONS= No-Null-Key,
 Compress-Key-Data(150)
*
idx-definition
*=============
NAME= IDX_CONTACT_ORG
ID= 4s
IDT-NAME= IDT_CONTACT
KEY-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("FIELD=Organization_Name KEY_LEVEL=Standard"),
 Field(Account),
 Null-Key("K$$$$$$$")
OPTIONS= No-Null-Key,
 Compress-Key-Data(150)
*
idx-definition
*=============
NAME= IDX_PROSPECT
ID= 5s
IDT-NAME= IDT_PROSPECT
KEY-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("FIELD=Person_Name KEY_LEVEL=Standard"),
 Field(Name),
 Null-Key("K$$$$$$$")
OPTIONS= No-Null-Key,
 Compress-Key-Data(150)
*
*
************************************************************************
* Loader and Job Definitions for Initial Load. You can remove the parameter
OPTIONS=APPEND, if you are not doing an incremental load
************************************************************************
*
loader-definition
*====================
NAME= All_Load
JOB-LIST= job-account,
 job-contact,
 job-prospect
*
loader-definition
*====================
NAME= siebel_prospect
JOB-LIST= job-prospect
OPTIONS= APPEND
*
loader-definition
*====================
NAME= siebel_contact
JOB-LIST= job-contact
OPTIONS= APPEND
*
loader-definition
```

```
*====================
NAME= siebel_account
JOB-LIST= job-account
OPTIONS= APPEND
*
job-definition
*==============
NAME= job-account
FILE= lf-input-account
IDX= IDX_ACCOUNT
*
job-definition
*==============
NAME= job-contact
FILE= lf-input-contact
IDX= IDX_CONTACT_NAME
OPTIONS= Load-All-Indexes
*
job-definition
*==============
NAME= job-prospect
FILE= lf-input-prospect
IDX= IDX_PROSPECT
*
*
logical-file-definition
*======================
NAME= lf-input-account
PHYSICAL-FILE= IDT_ACCOUNT
*PHYSICAL-FILE= "+/data/account.xml"
*************
* If Loading directly from Table, set PHYSICAL-FILE as Table Name,If loading from xml
file set PHYSICAL-FILE as XML file name
*************
INPUT-FORMAT= SQL
*FORMAT= XML
**********
*If Loading directly from Table, set INPUT-FORMAT as SQL, If loading from xml file use
INPUT-FORMAT as XML
*********
*
logical-file-definition
*======================
NAME= lf-input-contact
PHYSICAL-FILE= IDT_CONTACT
INPUT-FORMAT= SQL
*
logical-file-definition
*======================
NAME= lf-input-prospect
PHYSICAL-FILE= IDT_PROSPECT
INPUT-FORMAT= SQL
*
user-job-definition
*==================
COMMENT= "Load Accounts"
NAME= AccountLoad
*
user-step-definition
*====================
COMMENT= "Step 0 for acct load"
JOB= AccountLoad
NUMBER= 0
NAME= runAccountLoad
TYPE= "Load ID Table"
PARAMETERS= ("Loader Definition",siebel_account)
```

```
*
user-job-definition
*==================
COMMENT= "Load contacts"
NAME= ContactLoad
*
user-step-definition
*====================
COMMENT= "Load Contacts"
JOB= ContactLoad
NUMBER= 0
NAME= runContactLoad
TYPE= "Load ID Table"
PARAMETERS= ("Loader Definition",siebel_contact)
*
user-job-definition
*==================
COMMENT= "Load Prospects"
NAME= ProspectLoad
*
user-step-definition
*====================
COMMENT= "Step 0 for prospect load"
JOB= ProspectLoad
NUMBER= 0
NAME= runProspectLoad
TYPE= "Load ID Table"
PARAMETERS= ("Loader Definition",siebel_prospect)
*
search-definition
*================
NAME= "search-person-name"
IDX= IDX_CONTACT_NAME
COMMENT= "Use this to search and score on person"
KEY-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("FIELD=Person_Name SEARCH_LEVEL=Typical"),
 Field(Name)
SCORE-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("Purpose=Person_Name MATCH_LEVEL=Typical"),
 Matching-
Fields("Name:Person_Name,StreetAddress:Address_Part1,City:Address_part2,State:Attribut
e1,PrimaryPostalCode:Postal_area")
*
**********
* Depending on the Business requirement, you can add or remove the fields to be used for
matching from the "Matching-Fields" section
*********
search-definition
*================
NAME= "search-address"
IDX= IDX_CONTACT_ADDR
COMMENT= "Use this to search and score on person"
KEY-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("FIELD=Address_part1 SEARCH_LEVEL=Typical"),
 Field(StreetAddress)
SCORE-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("Purpose=Address MATCH_LEVEL=Typical"),
 Matching-Fields
```

```
("Name:Person_Name,StreetAddress:Address_Part1,City:Address_part2,State:Attribute1,Pri
maryPostalCode:Postal_area")
*
search-definition
*================
NAME= "search-company"
IDX= IDX_CONTACT_ORG
COMMENT= "Use this to search for a person within a company"
KEY-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("FIELD=Organization_Name SEARCH_LEVEL=Typical"),
 Field(Account)
SCORE-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("Purpose=Contact MATCH_LEVEL=Typical"),
 Matching-Fields
("Account:Organization_Name,Name:Person_Name,StreetAddress:Address_Part1")
*
search-definition
*================
NAME= "search-prospect-name"
IDX= IDX_PROSPECT
COMMENT= "Use this to search and score on prospect person"
KEY-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("FIELD=Person_Name SEARCH_LEVEL=Typical"),
 Field(Name)
SCORE-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("Purpose=Person_Name MATCH_LEVEL=Typical"),
 Matching-
Fields("Name:Person_Name,StreetAddress:Address_Part1,City:Address_Part2,State:Attribut
e1,PostalCode:Postal_Area")
*
search-definition
*================
NAME= "search-org"
IDX= IDX_ACCOUNT
COMMENT= "Use this to search and score on company"
KEY-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("FIELD=Organization_Name SEARCH_LEVEL=Typical"),
 Field(Name)
SCORE-LOGIC= SSA,
 System(default),
 Population(usa),
 Controls("Purpose=Organization MATCH_LEVEL=Typical"),
 Matching-Fields
("Name:Organization_Name,PAccountStrAddress:Address_Part1,PAccountCity:Address_Part2")
*
multi-search-definition
*======================
NAME= "multi-search-direct-contact"
SEARCH-LIST= "search-person-name,search-company,search-address"
IDT-NAME= IDT_CONTACT
*
multi-search-definition
*======================
NAME= "multi-search-contact"
SEARCH-LIST= "search-person-name,search-company"
IDT-NAME= IDT_CONTACT
```

```
*
multi-search-definition
*======================
NAME= "multi-search-person"
SEARCH-LIST= "search-person-name,search-address"
IDT-NAME= IDT_CONTACT
*
multi-search-definition
*======================
NAME= "multi-search-division"
SEARCH-LIST= "search-company,search-address"
IDT-NAME= IDT_CONTACT
*
Section: User-Source-Tables
*
**********************************************************************
* Initial Load Database Source Views
**********************************************************************
************************************
* Staging Table for Account Data
* Please refer the DQ Admin guide before changing the sequence of the fields
************************************
create_idt IDT_ACCOUNT
sourced_from odb:15:ssa_src/ssa_src@ISS_DSN
INIT_LOAD_ALL_ACCOUNTS.ACCOUNT_NAME Name V(100),
INIT_LOAD_ALL_ACCOUNTS.ACCOUNT_ADDR_ID DUNSNumber V(60),
INIT_LOAD_ALL_ACCOUNTS.ACCOUNT_ID (pk1) RowId C(30) ,
INIT_LOAD_ALL_ACCOUNTS.CITY PAccountCity V(100),
INIT_LOAD_ALL_ACCOUNTS.COUNTRY PAccountCountry V(60),
INIT_LOAD_ALL_ACCOUNTS.POSTAL_CODE PAccountPostalCode V(60),
INIT_LOAD_ALL_ACCOUNTS.STATE PAccountState V(20),
INIT_LOAD_ALL_ACCOUNTS.ADDRESS_LINE1 PAccountStrAddress V(100),
INIT_LOAD_ALL_ACCOUNTS.ACCOUNT_ADDR_ID (pk2) PAccountAddressID C(60)
SYNC REPLACE_DUPLICATE_PK
TXN-SOURCE NSA
;
**********************************************************************
* Sample entries if Loading the data from Flat File
**********************************************************************
*create_idt
* IDT_ACCOUNT
* sourced_from FLAT_FILE
* Name W(100),
* DUNSNumber W(60),
* PAccountCity W(100),
* PAccountCountry W(60),
* PAccountPostalCode W(60),
* PAccountState W(20),
* PAccountStrAddress W(100),
* (pk) RowId C(30)
*
*SYNC REPLACE_DUPLICATE_PK
*TXN-SOURCE NSA
*;
************************************
* Staging Table for Contact Data
************************************
create_idt IDT_CONTACT
sourced_from odb:15:ssa_src/ssa_src@ISS_DSN
INIT_LOAD_ALL_CONTACTS.BIRTHDATE BirthDate V(60),
INIT_LOAD_ALL_CONTACTS.CELLULARPHONE CellularPhone V(60),
INIT_LOAD_ALL_CONTACTS.EMAILADDRESS EmailAddress V(60),
INIT_LOAD_ALL_CONTACTS.NAME NAME V(100),
INIT_LOAD_ALL_CONTACTS.HOMEPHONE HomePhone V(60),
INIT_LOAD_ALL_CONTACTS.MIDDLE_NAME MiddleName V(100),
INIT_LOAD_ALL_CONTACTS.ACCOUNT Account V(100),
```

```
INIT_LOAD_ALL_CONTACTS.CONTACT_ID (pk1) RowId C(30),
INIT_LOAD_ALL_CONTACTS.SOCIALSECURITYNUMBER SocialSecurityNumber V(60),
INIT_LOAD_ALL_CONTACTS.WORKPHONE WorkPhone V(60) ,
INIT_LOAD_ALL_CONTACTS.CITY City V(60),
INIT_LOAD_ALL_CONTACTS.COUNTRY Country V(20),
INIT_LOAD_ALL_CONTACTS.POSTAL_CODE PrimaryPostalCode V(20),
INIT_LOAD_ALL_CONTACTS.STATE State V(20),
INIT_LOAD_ALL_CONTACTS.STREETADDRESS StreetAddress V(100),
INIT_LOAD_ALL_CONTACTS.ADDRESS_ID (pk2) ContactAddressID C(60)
SYNC REPLACE_DUPLICATE_PK
TXN-SOURCE NSA
;
************************************
* Staging Table for Prospect Data
************************************
create_idt IDT_PROSPECT
sourced_from odb:15:ssa_src/ssa_src@ISS_DSN
INIT_LOAD_ALL_PROSPECTS.ACCOUNT_NAME Account V(100),
INIT_LOAD_ALL_PROSPECTS.CELLULAR_PHONE CellularPhone V(60),
INIT_LOAD_ALL_PROSPECTS.CITY City V(60),
INIT_LOAD_ALL_PROSPECTS.COUNTRY Country V(30),
INIT_LOAD_ALL_PROSPECTS.EMAIL_ADDRESS EmailAddress V(60),
INIT_LOAD_ALL_PROSPECTS.NAME NAME V(100),
INIT_LOAD_ALL_PROSPECTS.HOME_PHONE HomePhone V(60),
INIT_LOAD_ALL_PROSPECTS.MIDDLE_NAME MiddleName V(100),
INIT_LOAD_ALL_PROSPECTS.POSTAL_CODE PostalCode V(20),
INIT_LOAD_ALL_PROSPECTS.SOCIAL_SECURITY_NUMBER SocialSecurityNumber V(60),
INIT_LOAD_ALL_PROSPECTS.STATE State V(20),
INIT_LOAD_ALL_PROSPECTS.STREETADDRESS StreetAddress V(100),
INIT_LOAD_ALL_PROSPECTS.WORK_PHONE WorkPhone V(100),
INIT_LOAD_ALL_PROSPECTS.PROSPECT_ID (pk) RowId C(30)
SYNC REPLACE_DUPLICATE_PK
TXN-SOURCE NSA
;
Section: Files
Section: Views
```

# 8 Administering Data Quality

## Administering Data Quality

This chapter explains how to administer data quality in order to perform your data matching and data cleansing tasks. It includes the following topics:

- *Data Quality Modes of Operation*
- *Data Quality Rules*
- *Data Quality Batch Job Parameters*
- *Cleansing Data Using Batch Jobs*
- *Matching Data Using Batch Jobs*
- *Merge Algorithm in the Object Manager Layer*
- *Merging Duplicate Records*
- *Process of Merging Duplicate Records*
- *Using Fuzzy Query*
- *Calling Data Matching and Data Cleansing from Scripts or Workflows*
- *Troubleshooting Data Quality*
- *Viewing Parameter and Field Mapping Values for Universal Connector*

## Data Quality Modes of Operation

Data cleansing and data matching operates in real-time or in batch mode.

In *real-time mode*, data quality functionality is called whenever a user attempts to save a new or modified account, contact, or prospective contact record to the database.

For data cleansing, the fields configured for data cleansing are standardized before the record is committed.

For data matching, when data quality detects a possible match with existing data, all probable matching candidates are displayed in real time. This helps to prevent duplication of records because:

- When entering data initially, users can select an existing record to continue their work, rather than create a new one.
- When modifying data, users can identify duplicates resulting from their changes.

In *batch mode*, you can use either the Administration - Server Management screen or the srvrmgr command-line utility to submit server component batch jobs. You run these batch jobs at intervals depending on business requirements and the amount of new and changed records.

For data cleansing, a batch run standardizes and corrects a number of account, contact, prospect, or business address fields. You can cleanse all of the records for a business component or a subset of records. For more information about data cleansing batch tasks, see *Cleansing Data Using Batch Jobs*.

For data matching, a batch run identifies potential duplicate record matches for account, contact, and prospect records. You can perform data matching for all of the records for a business component, or a subset of records. Potential duplicate records are presented to the data administrator for resolution in the Administration-Data Quality views. The duplicates can be resolved over time by a data steward (a person whose job is to monitor the quality of incoming and outgoing data for an organization.) For more information about data matching batch tasks, see *Matching Data Using Batch Jobs*.

# Real-Time Data Cleansing and Data Matching

In real-time mode, data quality is called when you save a new or modified record. If both data cleansing and data matching are enabled for the same object manager, data cleansing runs first.

If data cleansing is enabled, a set of fields preconfigured to use data cleansing are standardized before the record is committed.

If data matching is enabled, and the new record is a potential duplicate, one of the following dialog boxes appears:

- Duplicate Accounts
- Duplicate Contacts
- Duplicate Prospects

You must then decide the fate of the new record, as follows:

- If you think the record is not a duplicate, close the dialog box or click Ignore All.
  The new record remains saved in the database and no change takes place.
- If you think the record is a duplicate, select the best-matching record from the dialog box using the Pick button.
  The duplicate record that you choose becomes the surviving record and the new record gets deleted after a sequenced merge with the surviving record as described in *Sequenced Merges*.

In real-time mode, if you enter two new records that have the same Name and Location, then an error message displays similar to the following: *The same values for (Name, Location) already exist. To enter a new record, make sure that field values are unique.* Real-time data matching prevents creation of a duplicate record in the following ways:

- If you are in the process of creating a new record, that record is not saved.
- If you are in the process of modifying a record, the change is not made to the record.

> **Note:** Only certain fields are configured to support data matching and data cleansing. If you do not enter values in these fields when you create a new record, or you do not modify the values in these fields when changing a record, data cleansing and data matching are not triggered. For more information about which fields are preconfigured for different business components, see *Preconfigured Field Mappings for Oracle Data Quality Matching Server* and *Preconfigured Field Mappings for Oracle Data Quality Matching Server*.

# Batch Data Cleansing and Data Matching

Batch processing provides a means to cleanse and match a large number of records 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 in one of the following ways:

- Using the Siebel Server Manager command-line interface, 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. This is a convenient way of consolidating and reusing batch job parameters and also of overriding vendor parameters. For more information, see *Data Quality Rules*.

For more information about using the Siebel Server Manager and administering component jobs, see *Siebel System Administration Guide* . In particular, read the chapters about the Siebel Enterprise Server architecture, using the Siebel Server Manager GUI, and using the Siebel Server Manager command-line interface.

You must run batch mode key generation on all existing records before you run real-time data matching. The Universal Connector requires generated keys in the key tables first before you can run real-time data matching. The key generation is done within the deduplication task (which is the reason for running deduplication on all existing records first).

**CAUTION:** If you write custom Siebel CRM scripting on business components used for data matching (such as Account, Contact, List Mgmt Prospective Contact, and so on), the modifications to the fields by the script execute in the background and might not trigger logic that activates user interface features. For example, the scripting might not trigger UI features such as windows that show potential matching records.

# Data Quality Rules

In the Administration - Data Quality screen, Rules view, you can define rules for each of the data quality operations that are performed in real-time and in batch mode.

The data quality rules specify the parameters used when a data quality operation is performed in real-time or in batch mode. For example, you can create a rule for the batch mode Data Cleansing operation on the Account business component for a particular vendor. The parameters used are the vendor parameters defined for the applicable vendor, but you can override these parameters by specifying the equivalent rule parameters. However, the value set for Match Threshold in the User Preferences data quality settings override the equivalent rule parameters.

You can only create rules for business components for which data cleansing or data matching are supported. This includes the preconfigured business components and any additional business components that you configure for data cleansing and data matching. Also, you can only create rules for operations that are supported for a particular vendor. For each vendor, the supported operations and business components are defined in the Administration - Data Quality screen, Third Party Administration view.

You can create only one real time rule for each combination of vendor, business component, and operation name. However, you can create any number of batch rules for each combination of vendor business component, and operation name.

When you define a rule for real time mode, the rule is applied each time data cleansing or data matching is performed for the business component. When you define a rule for batch mode, the rule is applied if you specify the name of the rule in the batch job parameters, see *Data Quality Batch Job Parameters*. Using rules in this way allows you to consolidate batch job parameters into a reusable rule.

You can specify a search specification, business object name, business component name, threshold, and operation Type in a rule or in the job parameters when you submit a job in batch mode. The values in the job parameters override any value in the rules.

**Note:** Do not confuse data quality rules with the matching rules that are used by the third-party software.

# Creating a Data Quality Rule

Use the following procedure to create a data quality rule.

## To create a data quality rule

1. Navigate to the Administration - Data Quality screen, then the Rules view.
2. Create a new record. Some of the fields are shown in the following table:

| Field | Comment |
|---|---|
| Name | Enter a unique name for the rule. |
| Search Specification | Enter a search specification.<br><br>Applicable for Operation Type Batch only. |
| Vendor Name | Select a vendor name, for example, ISS. |
| Operation Type | Select Batch or Real Time. |
| Operation Name | Select one of the following:<br><br>o Data Cleansing<br>o DeDuplication<br>o Key Generate (batch mode only)<br>o Key Refresh (batch mode only) |
| Threshold | Enter a value between 50 and 100. This value overrides the value in the Data Quality settings.<br><br>Applicable for Operation Name DeDuplication only. |
| Source Business Component | Select a business component name. |
| Source Business Object | Select the business object name corresponding to the business components |

| Field | Comment |
|---|---|
| | |

An example of a rule is shown in the following table. This is a rule for DeDuplication operations for all Account records whose name starts with Aa.

| Field | Value |
|---|---|
| Name | Rule_Batch_Account_Dedup |
| Search Specification | [Name] LIKE 'Aa*' |
| Vendor Name | ISS |
| Operation Type | Batch |
| Operation Name | DeDuplication |
| Threshold | 60 |
| Source Business Component | Account |
| Source Business Object | Account |

3. (Optional) Specify rule parameters.
   a. Click the Rule Parameter view tab.
   b. Create rule parameters by selecting a parameter and entering the required value.

# Data Quality Batch Job Parameters

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 |
|---|---|---|
| Buscomp name<br><br>bcname | Yes | The name of the business component: Possible values include:<br>• Account |

| Job Parameter or Server Manager Parameter | Required | Description |
|---|---|---|
| | | • Contact<br>• List Mgmt Prospective Contact<br>• Business Address - applicable to Data Cleansing operations only. For Siebel Industry Applications, CUT Address is used instead of Business Address. |
| Business Object Name<br><br>bobjname | Yes | The name of the business object. Possible values include:<br>• Account<br>• Contact<br>• List Mgmt<br>• Business Address - applicable to Data Cleansing operations only |
| Operation Type<br><br>opType | Yes | The type of operation: Possible values are:<br>• Data Cleansing - cleanses data<br>• Key Generate - generates match keys<br>• Key Refresh - refreshes match keys<br>• DeDuplication - performs data matching. |
| Object Sorting Clause<br><br>objsortclause | No | Applicable to Data Matching operations only.<br><br>Indicates how candidate records are sorted for optimal processing by the data matching software. The default value is Dedup Token. |
| Object Where Clause<br><br>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 up large data quality batch tasks using the first letter of the name.<br><br>For example, the following object WHERE clause selects only French account records where the account name begins with A:<br><br>`[Name] like 'A*' AND [Country] = 'France'`<br><br>As another example, the following object WHERE clause selects all records where Name begins with Paris or ends with london:<br><br>`[Name] like 'Paris*' or [Name] like '*london'` |
| Data Quality Setting<br><br>DQSetting | No | Specifies data quality settings for data cleansing and data matching jobs. This parameter has three values separated by commas:<br>• **First value.** If this value is set to Delete, existing duplicates are deleted. Otherwise, existing duplicates are not deleted. This is the only usage for this value.<br>• **Second value.** Applicable to the Universal Connector only. It specifies whether the job is a full or incremental data matching job.<br>• **Third value.** This is obsolete. Enter an empty string.<br>For more information about the use of DQSetting, see *Matching Data Using Batch Jobs*. |
| Threshold | No | Specifies a value for the Threshold data quality parameter. |

| Job Parameter or Server Manager Parameter | Required | Description |
|---|---|---|
| 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, Rules view. For example:<br>`RuleName="Rule_Batch_Account_Dedup"`<br>For more information, see *Data Quality Rules*. |

# 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'
```

**CAUTION:** When you run a process in batch mode, any visibility limitation against your targeted data set is ignored. It is recommended that you allow only a small group of people to access the Siebel Server Manager to run your data quality tasks, otherwise you run the risk of corrupting your data.

## To perform batch mode data cleansing

1. Start the Server Manager Program.
2. At the srvrmgr prompt, enter a command like one of those in the following table to perform data cleansing.

| Business Component | Example of Server Manager Command |
|---|---|
| Account | run task for comp DQMgr with bcname=Account, bobjname= Account, opType="Data Cleansing", objwhereclause="[*field_name*] LIKE '*search_string**'", DqSetting="","","" |
| 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'" |

# Matching Data Using Batch Jobs

Depending on your business requirements, you might want to use batch jobs to perform data matching on some or all of the records in the supported business components. If you run a data matching batch job on all the records in a business component, the work can often be completed more quickly by splitting the work into a number of smaller batch jobs (not more than 50,000 to 75,000 records at a time). When data matching has been performed on all of the records in the business component, you can run future data matching batch jobs on just the new or changed records.

If you want to perform data matching for some number of mutually-exclusive subsets of the records in a business component, such as all the records where a field name starts with a given letter, use a separate job to specify each subset, with WHERE clauses as follows:

```
objwhereclause="[field_name] LIKE 'A*'"
objwhereclause="[field_name] LIKE 'B*'"
...
objwhereclause="[field_name] LIKE 'Z*'"
objwhereclause="[field_name] LIKE 'a*'"
...
objwhereclause="[field_name] LIKE 'z*'"
```

The following example further describes batch data matching.

# Example of Batch Data Matching Using the Universal Connector

You must run batch mode key generation on all existing records before you run real-time data matching. The Universal Connector requires generated keys in the key tables first before you can run real-time data matching. The key generation is done within the deduplication task, which is the reason for running deduplication on all existing records first. For more information about batch data cleansing and matching, see *Batch Data Cleansing and Data Matching*.

The following procedure describes how to start a data matching batch job.

## To perform batch mode data matching

1. Follow the instructions in *Generating or Refreshing Keys Using Batch Jobs*.
2. At the srvrmgr prompt, enter commands like those in the following table to perform data matching.

| Business Component | Example of Server Manager Command |
|---|---|
| Account | run task for comp DQMgr with DqSetting="'Delete'", bcname=Account, bobjname=Account, opType=DeDuplication, objwhereclause="[Name] like '*search_string*'" |
| Contact | run task for comp DQMgr with DqSetting="'Delete'", bcname=Contact, bobjname=Contact, opType=DeDuplication, objwhereclause="[Name] like '*search_string*'" |
| List Mgmt Prospective Contact | run task for comp DQMgr with DqSetting="'Delete'", bcname="List Mgmt Prospective Contact", bobjname="List Mgmt", opType=DeDuplication, objwhereclause="[Name] like '*search_string*'" |

## Full Data Matching Jobs

In a full data matching job, the records for which you want to locate duplicates and the candidate records that can include those duplicates are defined by the same search specification. A full data matching job is specified with the value Yes in the DQSetting parameter, see the following table.

Full data matching jobs are useful when:

- You want to perform data matching on a whole database table.
- You are setting up the data quality installation.
- You perform data matching for the customer data for a particular business component for the first time.

A typical example of a command for a full data matching job is as follows:

```
run task for comp DQMgr with DqSetting="'','Yes','account_match.xml'",
bcname=Account, bobjname=Account, opType=DeDuplication, objwhereclause="[Name] LIKE
'A*'"
```

Jobs like this that perform data matching for a subset of records are still considered to be full data matching jobs because the data to be checked does not depend on earlier data matching.

## Incremental Data Matching Jobs

If you want to perform data matching for some number of nonexclusive subsets of the records in a business component, such as all the records that have been created or updated since you last ran data matching, use a WHERE clause that includes an appropriate timestamp, and also adjust the DqSetting clause of the command as shown in the following table.

| DqSetting Parameter Sequence | Valid Values | Comments |
|---|---|---|
| First section | Leave blank | Specify as two adjacent quotation marks. |
| Second section<br><br>(Enforce Search Spec on Candidate Records) | Yes or No (default) | Specifies whether or not the same search specification is used for both the records whose duplicates are of interest and the candidate records that can include those duplicates. Use Yes for full data matching batch jobs. Use No for incremental data matching batch jobs. |
| Third section | Leave blank | None. |

This kind of job is considered an *incremental data matching* job because data matching was done earlier and does not have to be redone at this time. In an incremental data matching batch job, the records for which you want to locate duplicates are defined by the search specification, but the candidate records that can include those duplicates can be drawn from the whole applicable database table. Incremental data matching batch jobs are useful if you run them regularly, such as once a week. A typical example of a command for an incremental data matching job is as follows:

```
run task for comp DQMgr with DqSetting="'','No',''",
bcname=Account, bobjname=Account, opType=DeDuplication, objwhereclause="[Updated] >
'08/18/2005 20:00:00'
```

**Note:** If you do not specify the DQSetting parameter, or leave the second value of the DQSetting parameter blank, the job will be an incremental data matching job.

# Generating or Refreshing Keys Using Batch Jobs

The following procedure describes how to start a batch job to generate or refresh keys for data matching.

## To start a batch job to generate or refresh keys

1. Start the Server Manager Program.
2. At the srvrmgr prompt, enter one of the commands in the following table to generate or refresh keys.

   Substitute values of your own choosing in the WHERE clauses, as needed.

| Business Component | Generate or Refresh Keys? | Example of Server Manager Command |
|---|---|---|
| Account | Generate | run task for comp DQMgr with bcname=Account, bobjname=Account, opType="Key Generate", objwhereclause="[Updated] > '07/18/2005 16:00:00'" |
| Account | Refresh | run task for comp DQMgr with bcname=Account, bobjname=Account, opType="Key Refresh", objwhereclause="[Name] LIKE '*search_string*'" |
| Contact | Generate | run task for comp DQMgr with bcname=Contact, bobjname=Contact, opType="Key Generate", objwhereclause="[Updated] > '07/01/2005 14:10:00'" |
| Contact | Refresh | run task for comp DQMgr with bcname=Contact, bobjname=Contact, opType="Key Refresh", objwhereclause="[Last Name] LIKE '*search_string*'" |
| List Mgmt Prospective Contact | Generate | run task for comp DQMgr with bcname="List Mgmt Prospective Contact", bobjname="List Mgmt", opType="Key Generate", objwhereclause="[Updated] > '07/18/2005 16:00:00'" |
| List Mgmt Prospective Contact | Refresh | run task for comp DQMgr with bcname="List Mgmt Prospective Contact", bobjname="List Mgmt", opType="Key Refresh", objwhereclause="[Last Name] LIKE '*search_string*'" |

   The examples in the table show slightly different WHERE clauses for key generation and key refresh operations, as follows:

   - The generation commands generate keys for all records in the business component that have been updated since the specified date and time.

- The refresh commands refresh keys for all records in the business component that match the search string in the specified field.

You can use either of these two types of WHERE clauses for both generation and refresh operations.

If you want to generate or refresh keys for all records in the business component, use a WHERE clause containing a wildcard character (*) to match all records, as follows:

```
objwhereclause="[
field_name] LIKE '*'"
```

# Customizing Data Quality Server Component Jobs for Batch Mode

Rather than specifying parameters each time you start a data quality batch job, you can customize the Data Quality Manager server component with the parameters that you require. This is mainly for ease of use when starting tasks using the srvrmgr program.

You use the Administration - Server Configuration views to create customized components according to the Data Quality Manager Server component. You specify Data Quality Manager as the Component Type.

- For more information about creating custom component definitions, see *Siebel System Administration Guide* .
- For information about component customization for your third-party data quality vendor software, consult your third-party vendor.

You must enable new custom Data Quality Manager components before you can use them. And, if you change parameters of running components, you must shut down and restart the components or restart the Siebel Server for the changes to take effect.

> **Note:** For Siebel CRM Version 7.8 and later, you can also set specific parameters for a data quality task and save the configuration as a template by using the Administration - Server Configuration screen, Job Templates view. The benefit in doing so is that there is no need to copy component definitions. For more information about Siebel CRM templates, see *Configuring Siebel Business Applications* .

# Merge Algorithm in the Object Manager Layer

The Merge Records functionality is used by customers to enhance data quality. For example, duplicate accounts might be merged to a target account or you might want to merge duplicate opportunities. To call the feature, select two or more records and choose Edit, then Merge Records from the application-level menu. For more information about the Merge Records menu option, see *Siebel Fundamentals* .

# Example of the Merge Records Process

You want to merge accounts A1 and A2 into A1. These accounts might have child quotation marks or associated contacts. These relationships are defined using one-to-many or many-to-many links. The following describes what happens after the merge:

- Account A2 is deleted after the merge.
- The contacts associated with A2 are associated with A1 after the merge.

The links defined between the business components are used to implement the merge algorithm. The algorithm used by the merge process at the OM layer is explained in the following topics for one-to-many and many-to-many links.

# Overview of Merge Algorithm

The following section provides a brief overview of what happens during the merge process using the example of merging accounts A1 and A2 into A1.

The merge process starts by enumerating through all link definitions that might be relevant, for example, in the case of the example, where the source business component is accounts.

## One-to-Many Relationship

A one-to-many relationship defines the destination field, which is the foreign key in the detail table that points to a row in the parent table. Only links where the source field is "Id", that is, where the foreign key in the detail table stores the ROW_ID of the parent table row, are considered.

To make children of A2 point to A1, the merge must update the destination field in the detail table to now point to the ROW_ID of A1.

**User property name:** Use Literals For Merge

**Use Literals for Merge:** S_BU

**Value:** TRUE

When merging two records, the child records of the loser record point to the remaining record and the LAST_UPD and LAST_UPD_By columns of those child records are also updated. For example, account A2 is merged to account A1. Account A2 has service request SR1, and SR2. The columns LAST_UPD, and LAST_UPD_BY of SR1 and SR2 are updated during merge process.

From the example, link account or quote foreign key in S_DOC_Quote is account Id (TARGET_OU_ID). TARGET_OU_ID stored the ROW_ID of the A2. It is now updated to point to ROW_ID of A1.

SQL generated:

```
UPDATE S_DOC_QUOTE set TARGET_OU_ID = 'Row Id of A1'
```

where:

TARGET_OU_ID is equal to 'Row Id of A2'

While the merge is processing the link account or quote, it also checks to see whether there are other foreign keys from quote pointing to account using the join definitions. These keys are also updated.

An optimization is used to ensure that there are no redundant update statements. For example, if there are two links defined (account or quote and account or quote with primary with the same destination field Account Id), the process would update TARGET_OU_ID of S_DOC_QUOTE twice to point to A1. To avoid this scenario, a map of table name or column name of the processed field is maintained. The update is skipped if the column has been processed before.

After the update you might have duplicate children for an account. For example, if the unique key for a quote is the name of the quote, merging two accounts with quotation marks of the same name will result in duplicates. The CONFLICT_ID column of children that will become duplicates after the merge is updated. This operation is performed before the actual update.

The user must examine duplicate children (identified by CONFLICT_ID being set) to make sure that they are true duplicates. For example, if the merged account has child quotation marks named Q1 and Q1, it is possible that these refer to distinct quotation marks. If this is the case, the name of one of the quotation marks must be updated and the children must be merged.

### Many-to-Many Relationship

The many-to-many relationship (Accounts-Contacts) differs slightly from the one-to-many relationship in that it is implemented using an intersection table that stores the ROW_IDs of parent-child records. On a merge, the associations must be updated. The Contacts associated with the old Account is now associated with the new Account.

The Inter parent column of the intersection table is updated to point to the new parent. As in the one-to-many case, to avoid redundant updates, a map of intersection tables that have been processed is maintained. Therefore, if the source and target business components use the same base table, both child and parent columns are updated.

The CONFLICT_ID column of intersection table entries that become duplicates after the merge is updated.

In contrast to the one-to-many link case, duplicates in the intersection table imply that the same child is being associated with the parent two or more times. However, there might be cases where the intersection table has entries besides the ROW_ID of the parent and child rows that store information specific to the association.

The duplicate association records are only preserved when records are determined as unique (according to the intersection table unique key). This means those duplicate association records might have some unique attributes and these attributes are part of a unique key of the intersection table. CONFLICT_ID does not account for uniqueness among records.

## Merging Duplicate Records

After you run data matching in batch mode, duplicate records are displayed in the Duplicate Accounts, Duplicate Contacts, and Duplicate Prospects views in the Administration - Data Quality screen. You can then determine which records you want to retain and which records you want to merge with the retained record.

> **CAUTION:** Merging records is an irreversible operation. You must review all records carefully before using the following procedure and initiating a merge.

You can merge duplicate records in the following ways:

- **Merge Records option (Edit, Merge Records).** Performs the standard merge functionality available in Siebel CRM for merging records. That is, this action keeps the record you indicate and associates all child records from

the nonsurviving record to it before deleting the nonsurviving record. For more information about the Merge Records menu option, see *Siebel Fundamentals* .

- **Merge button (from appropriate Duplicate Resolution View).** Performs a *sequenced merge* of the records selected in the sequence specified. This includes populating currently empty fields in the surviving record with values from the nonsurviving records, as described in *Sequenced Merges*. This action also performs a cleanup in the appropriate Deduplication Results table to remove the unnecessary duplicate records. This is the preferred method for deduplicating account, contact, and prospect records.

## Sequenced Merges

You use a sequenced merge to merge multiple records into one record. You assign sequence numbers to the records so that the record with the lowest sequence number becomes the surviving record, and the other records, the nonsurviving records, are merged with the surviving record.

When records are merged using a sequence merge, the following rules apply:

- All non-NULL fields from the surviving record are kept.

  Any fields that were NULL in the surviving record are populated by information (if any) from the nonsurviving records. Missing fields in the surviving record are populated in ascending sequence number order from corresponding fields in the nonsurviving records.
- The children and grandchildren (for example, activities, orders, assets, service requests, and so on) of the nonsurviving records are merged by associating them with the surviving record.

Sequenced merge is especially useful if many fields are empty, such as when a contact record with a Sequence of 2 has a value for Email address, but its Work Phone # field is empty, and a contact record with a Sequence number of 3 has a value of Work Phone #. If the field Email address and Work Phone # in the surviving record (sequence number 1) are empty, the value of Email address is taken from the records with sequence number 2, and the value of Work Phone # is taken from the record of sequence number 3.

A sequence number is required for each record even if there are only two records.

## Field Characteristics for Sequenced Merges

A field must have the following specific characteristics to be eligible for use in a sequenced merge:

- The field cannot be a calculated field and must reside on a physical database column.
- The field must be active, that is designated as Active in the respective business component.

# Process of Merging Duplicate Records

When you run a batch process, and depending on the number of duplicates in your system, you might find there are hundreds of rows in the Duplicate Accounts, Duplicate Contacts, and Duplicate Prospect views (in the Administration

- Data Quality screen). In this case, it is recommended that you use the following process to filter and merge duplicate records:

1. *Filtering Duplicate Records*

   This involves creating a query to find a subset of the duplicate records and then review the query results. For example, you might want to create a query that includes a subset of all duplicate records where the Name field starts with the letter A.
2. *Merging Duplicate Records*

   After the query results appear, you merge duplicate records using either the Merge button or the Merge Records option.

**CAUTION:** You must perform batch data matching first before trying to resolve duplicate records. For more information about batch data matching, see *Batch Data Cleansing and Data Matching*.

# Filtering Duplicate Records

Use the following procedure to filter duplicate records. This task is a step in *Process of Merging Duplicate Records*.

**Note:** You can use either standard or fuzzy query methods, depending on your needs. For more information about using fuzzy query, see *Using Fuzzy Query*.

## To filter duplicate records

1. Navigate to the Administration - Data Quality screen in your Siebel application.
2. Click one of the following links:
   - Duplicate Accounts
   - Duplicate Contacts
   - Duplicate Prospects
3. Click Query, enter your search criteria, and then click Go.

   The search results appear.

You now decide what you want to do with the duplicate records.

# Merging Duplicate Records

You must follow a slightly different procedure to merge child duplicate records. If you do not follow the correct procedure, orphan records can be created. Use the following procedure to merge duplicate records. This task is a step in *Process of Merging Duplicate Records*.

## To merge duplicate records

1. In the Administration - Data Quality screen, click the Duplicate *XXX* view for the type of record you have selected, where *XXX* is either Accounts, Contacts, or Prospects.

   For example, click the Duplicate Accounts view.

2. In the Duplicate view, drill down on one of the duplicate records.
   The appropriate Duplicate *XXX* Resolution view appears. The child applet shows the list of duplicate rows with the parent record appearing as the first row.
3. If two or more records appear to be duplicates, enter a sequence number in the Sequence field for each record.
4. Edit the records, if necessary.
   For example, you might want to keep some values from fields in nonsurviving records. In this case, you can make fields NULL in what will be the surviving records. The values from the corresponding fields in the nonsurviving records are then used to populate the NULL fields after the sequenced merge.
5. Select the records to be merged.
6. Click Merge.
   The records are merged to produce one new record. The record with the lowest sequence number assigned is retained after the merge. Missing fields in the retained record are populated from corresponding fields in the nonsurviving records, as described in *Sequenced Merges*.

## Merging Child Duplicate Records

Use the following procedure to merge child duplicate records.

### To merge child duplicate records

1. In the appropriate Duplicate *XXX* Resolution view, enter 1 in the Sequence field for the parent record.
2. Enter 2 and so on in the Sequence field for each of the child duplicate records.
3. Select the records to be merged, and select the parent records last.
4. Click Merge.

# Using Fuzzy Query

To run a query using fuzzy query, this facility must be enabled and several conditions must be met as described in *Enabling and Disabling Fuzzy Query*.

In particular:

- The query must not use wildcards.
- The query must specify values in fields designated as fuzzy query mandatory fields. For information about identifying the mandatory fields, see *Identifying Mandatory Fields for Fuzzy Query*.
- The query must leave optional fields blank.

If the conditions for fuzzy query are not satisfied, then any queries you make use standard query functionality.

## Using Fuzzy Query for Accounts

Use the following procedure to use fuzzy query for accounts.

### To use fuzzy query for accounts

1. Navigate to Accounts screen, then the Accounts List view.

2. Click the Query button.
3. Enter your query, and then click Go.

The query results contain fuzzy matches in addition to regular query matches.

# Using Fuzzy Query for Contacts

Use the following procedure to use fuzzy query for contacts.

## To use fuzzy query for contacts

1. Navigate to Contacts screen, then the Contacts List view.
2. Click the Query button.
3. Enter your query, and then click Go.

   The query results contain fuzzy matches in addition to regular query matches.

## Using Fuzzy Query for Prospects

Use the following procedure to use fuzzy query for prospects.

### To use fuzzy query for prospects

1. Navigate to List Management screen, then the Prospects view.
2. Click the Query button.
3. Enter your query, and then click Go.

   The query results contain fuzzy matches in addition to regular query matches.

## Example of Enabling and Using Fuzzy Query with Accounts

This topic gives an example of enabling and using fuzzy query. You might use this feature differently, depending on your business model.

In the following example, you enable fuzzy query for accounts, and then enter the query criteria. The query results contain fuzzy matches from the DeDuplication business service in addition to regular query matches.

**Note:** EAI Siebel Adapter does not support fuzzy queries. In addition, scripting does not support fuzzy queries.

### To enable and use fuzzy query for accounts

1. Perform the steps in *Enabling and Disabling Fuzzy Query*.
2. Perform the steps in *Enabling Data Quality at the User Level*.

   **Note:** For this example, set the Fuzzy Query - Max Returned data quality setting to 10.

3. Navigate to the Accounts screen, then the Account list view.
4. Enter your query, and then click Go.

   For this example, in the Name field, enter `Symphony`.

   Up to 10 records having Name set to Symphony are displayed.

> **Note:** If the number of Symphony account records is fewer than 10, then the fuzzy query results includes records where symphony is lowercase (as well as uppercase). For example, if four records for *Symphony* and 100 records for *symphony* are found in the database, the fuzzy query result shows four *Symphony* records and six *symphony* records. However, if fuzzy query is disabled, only the four *Symphony* records appear.

# Calling Data Matching and Data Cleansing from Scripts or Workflows

This topic provides information about calling data matching and data cleansing methods from external callers such as scripts or workflows.

You can call data quality from external callers to perform data matching. You can use the Value Match method of the Deduplication business service to:

- Match data in field or value pairs against the data within Siebel business components
- Prevent duplicate data from getting into the Siebel application through non-UI data streams

You can also call data quality from external callers to perform data cleansing. There are preconfigured Data Cleansing business service methods—Get Siebel Fields and Parse. Using an external caller, such as scripting or a workflow process, you first call the Get Siebel Fields method, and then call the Parse method to cleanse contacts and accounts.

The following scenarios provide more information about calling data quality from external callers:

- *Scenario for Data Matching Using the Value Match Method*
- *Scenario for Data Cleansing Using Data Cleansing Business Service Methods*

## Scenario for Data Matching Using the Value Match Method

This topic gives one example of how you can call the Value Match business service method using Siebel Workflow. You can use the Value Match method differently, depending on your business model.

In this scenario, a company must add contacts into the Siebel application from another application in the enterprise. To avoid introducing duplicate contacts into the Siebel application, the implementation uses a workflow process that includes steps that call EAI adapters and a step that calls the Value Match method.

In this case, the implementation calls the Value Match method as a step in the workflow process that adds the contact. This step matches incoming contact information against the contacts within the Siebel database. To prevent the introduction of duplicate information into the Siebel application, the implementation adds processing logic to the script using the results returned in the Match Info property set. The company can either reject potential duplicates with a high score, or it can include additional steps to add likely duplicates as records in the DeDuplication Results Business Component, so that they immediately become visible in the appropriate Duplicate *Record* Resolution view.

For information about how to call and use the Value Match method, see *Value Match Method*.

# Scenario for Data Cleansing Using Data Cleansing Business Service Methods

This topic gives one example of how you can call the Data Cleansing business service methods using Siebel Workflow. You might use the methods differently, depending on your business mode.

A system administrator or data steward in an enterprise wants to cleanse data before it enters the data through EAI or EIM interfaces. To do this, the system administrator or data steward uses a script or workflow that cleanses the data. The script or workflow calls the Get Siebel Fields method, which returns a list of cleansed fields for the applicable business component. Then the script or workflow calls the Parse method, which returns the data for the cleansed fields. For information about how to call and use the Get Siebel Fields and Parse methods, see *Data Cleansing Business Service Methods*.

# Deduplication Business Service Methods

This topic describes the following Deduplication business service method: *Value Match Method*. *Scenario for Data Matching Using the Value Match Method* gives one example of how you can call the Deduplication business service Value Match method.

> **Note:** For information about other deduplication business service methods that are available, see *Siebel Tools Online Help* .

## Value Match Method

You can use the Value Match method of the Deduplication business service to find potential matching records in the Siebel application or when you want to prevent duplicate data from getting into the Siebel application through non-UI data streams. For more information about business services and methods, see *Siebel Developer's Reference* .

### Arguments

The Value Match method consists of input and output arguments, some of which are property sets. The following table describes the input arguments, and the next table describes the output arguments.

> **CAUTION:** The Value Match method arguments are specialized. Do not configure these components.

| Name | Type | Property Name | Description | Comments |
|---|---|---|---|---|
| Adapter Settings | Property Set | Threshold | The threshold score for a duplicate record. A match is considered only if the score exceeds this value. | Optional. The value Override can be specified to override the corresponding setting information obtained by the service from the administration screens, vendor properties, and so on. |

| Name | Type | Property Name | Description | Comments |
|---|---|---|---|---|
| Match Values | Property Set | *Business component field names, and value* pairs:<br><br><Name1><Value1>,<br><Name2><Value2>,<br><Name3><Value3>, ... | The matched business component's field name and the corresponding field value:<br><br>`(Last Name, 'Smith')`<br>`(First Name, 'John'),`<br>`and so on ...`<br><br>**Note:** Each pair must be a child property set of Match Values. | These name-value pairs are used as the matched value rather than the current row ID of the matched business component. The vendor field mappings for the matched business component are used to map the business component field names to vendor field names. |
| BC Name | Property | BC Name | The name of the matched business component. | Required. |
| Update Modification Date | Property | Update Modification Date | If set to N, the match modification date is not updated. | Optional. The default is Y. |
| Use Result Table | Property | Use Result Table | If set to N, matches are not added to the result table. Instead, matches are determined by the business service. | Optional. The default is Y. |
| MultipleChild EntitiesDedup | Property | Support Multiple Child Entities Dedup | If the input argument is True, the data in the Match Values input argument is in a new data format.<br><br>**Note:** To support the Multiple Child Entities Dedup feature, vendors must support hierarchical data format. | Required. |

**Note:** Adapter Settings and Match Values are child property sets of the input property set.

## Return Value

For each match, a separate child property set called Match Info is returned in the output with properties specific to the match (such as Matchee Row ID and Score), as well as some general output parameters as shown in the following table.

**CAUTION:** The Value Match method arguments are specialized. Do not configure these components.

| Name | Type | Property Name | Description | Comments |
|---|---|---|---|---|
| End Time | Property | End Time | The run end time. | None |
| Match Info<br><br>**Note:** Match Info is a child property set of the output property set. | Property Set | Matchee Row ID | The row ID of a matching record. | If you match against existing records, the record ROW_IDs are found and returned in the Match Info property set. |
| | | Score | The score of a matching record. | |
| Num Results | Property | Num Results | The number of actual matches. | None |
| Start Time | Property | Start Time | The run start time. | None |

## Called From

Any means by which you can call business service methods, such as with Siebel e Script or from a workflow process.

## Example

The following is an example of using Siebel eScript to call the Value Match method. This script calls the Value Match method to look for duplicates of John Smith from the Contact business component and then returns matches, if any. After the script finishes, determine what you want to do with the duplicate records, that is, either merge or remove them.

```
function Script_Open ()
{

TheApplication().TraceOff();
TheApplication().TraceOn("sdq.log", "Allocation", "All");
TheApplication().Trace("Start of Trace");

// Create the Input property set and a placeholder for the Output property set
var svcs;
var sInput, sOutput, sAdapter, sMatchValues;
var buscomp;

svcs = TheApplication().GetService("DeDuplication");
sInput = TheApplication().NewPropertySet();
sOutput = TheApplication().NewPropertySet();
sAdapter = TheApplication().NewPropertySet();
sMatchValues = TheApplication().NewPropertySet();

// Set Generic Settings input property parameters
sInput.SetProperty("BC Name", "Contact");
sInput.SetProperty("Use Result Table", "N");
sInput.SetType("Generic Settings");

// Set Match Values child input property parameters
sMatchValues.SetProperty("Last Name", "Smith");
sMatchValues.SetProperty("First Name", "John");
sMatchValues.SetType("Match Values");
sInput.AddChild(sMatchValues);

// Set Adapter Settings child input property parameters
```

```
sAdapter.SetProperty("Search Level", "Narrow");
sAdapter.SetProperty("Population", "Default");
sAdapter.SetType("Adapter Settings");
sInput.AddChild(sAdapter);

// Invoke the "Value Match" business service
TheApplication().Trace("Property set created, ready to call Match method");
svcs.InvokeMethod("Value Match", sInput, sOutput);

// Get the Output property set and its values
TheApplication().Trace("Value Match method invoked");
var propName = "";
var propVal = "";
propName = sOutput.GetFirstProperty();
while (propName != "")
{
 propVal = sOutput.GetProperty(propName);
 TheApplication().Trace(propName);
 TheApplication().Trace(propVal);
 propName = sOutput.GetNextProperty()

 }
 TheApplication().Trace("End Of Trace");
 TheApplication().TraceOff();

}
```

# Data Cleansing Business Service Methods

This topic describes the following data cleansing business service methods: *Get Siebel Fields Method* and *Parse Method*. *Scenario for Data Cleansing Using Data Cleansing Business Service Methods* gives one example of how you can call the data cleansing business service methods.

## Get Siebel Fields Method

Get Siebel Fields is one of the methods of the Data Cleansing business service. This method returns a list of cleansed fields for a given business component. For more information about business services and methods, see *Siebel Developer's Reference* .

### Arguments

Get Siebel Fields arguments are listed in the following table.

| Argument Name | Display Name | Input or Output | Data Type | Description | Required? |
|---|---|---|---|---|---|
| BusComp Name | Bus Comp Name | Input | String | The name of the business component. | No |
| Field Names | Field Names | Output | Hierarchy | The name of the hierarchy. | Yes |

### Return Value

Child values: Name of the properties are Field 1, Field 2, and so on and corresponding values are Field Name.

### Usage

This method is used with the Parse method in the process of cleansing data in real time, and it is used with the Parse All function in the process of using a batch job to cleanse data.

### Called From

Any means by which you can call business service methods, such as with Siebel Workflow or Siebel eScript.

## Parse Method

Parse is one of the methods of the Data Cleansing business service. This method returns the cleansed field data. For more information about business services and methods, see *Siebel Developer's Reference* .

### Arguments

Parse arguments are listed in the following table.

| Argument Name | Display Name | Input or Output | Data Type | Description | Required? |
|---|---|---|---|---|---|
| BusComp Name | Bus Comp Name | Input | String | The name of the business component. | No |
| Input Field Values | Input Field Values | Input | Hierarchy | A list of field values. | Yes |
| Output Field Values | Output Field Values | Output | Hierarchy | A list of field values. | Yes |

### Return Value

Child name values are Field Name and Field Date.

### Usage

This method is used following the Get Siebel Fields method in the process of cleansing data in real time.

### Called From

Any means by which you can call business service methods, such as with Siebel Workflow or Siebel eScript. For more information about Siebel Workflow, see *Siebel Business Process Framework: Workflow Guide* .

# Troubleshooting Data Quality

If data cleansing or data matching is not working properly in real-time mode, check the following:

- **License key.** Verify that your license keys include data quality functionality.
- **Application object manager configuration** Verify that data cleansing or data matching has been enabled for the application you are logged into. For more information, see *Levels of Enabling and Disabling Data Cleansing and Data Matching* and *Specifying Data Quality Settings*.

- **User Preferences.** Verify that data cleansing or data matching has been enabled for the user. For more information, see *Enabling Data Quality at the User Level*.
- **Third-party software.** Verify that the third-party software is installed and you have followed all instructions from the third-party installation documents.

If you have configured new business components for data cleansing or data matching, also check the following:

- **Business component Class property.** Verify that the business component Class property is CSSBCBase.
- **Vendor Properties.** Verify that the vendor parameters and vendor field mappings have the correct values and that the values are formatted correctly. For example, there must be a space after a comma in vendor properties that have a compound value.

**Tip:** Check My Oracle Support regularly for updates to troubleshooting and other important information. For more information, see *Information about Data Quality on My Oracle Support*.

# Viewing Parameter and Field Mapping Values for Universal Connector

Universal Connector definitions are configured as vendor parameters in the Administration - Data Quality screen, Third Party Administration view. Use the following procedure to access and view preconfigured vendor parameters.

## To view preconfigured vendor parameters

1. Navigate to the Administration - Data Quality screen, then the Third Party Administration view.
2. In the Vendor list, select the record with, for example, the name Informatica Identity Resolution.
3. Click the Vendor Parameter view tab.

   The vendor parameters are displayed in the Vendor Parameters list. You must not reconfigure the parameter settings.

The field mappings from vendor fields to Siebel application fields are configured in field mapping parameters in the Administration - Data Quality screen, Third Party Administration view. There are field mappings for each of the supported business components and operations. Use the following procedure to access and view preconfigured field mappings.

# Viewing Preconfigured Field Mappings

Use the following task to view preconfigured field mappings.

## To view preconfigured field mappings

1. Navigate to the Administration - Data Quality screen, then the Third Party Administration view.
2. In the Vendor List, select the record with, for example, the name Informatica Identity Resolution.
3. Click the BC Vendor Field Mapping view tab.
4. In the BC Operation list, select the record for the required business component and operation.

   The field mappings are displayed in the Field Mapping list.

## Related Topics

*Configuring Vendor Parameters*

*Mapping of Vendor Fields to Business Component Fields*

# 9 Related Data Quality Objects in Siebel Tools

## Related Data Quality Objects in Siebel Tools

Siebel Data Quality (SDQ) is a user based license, containing the underlying infrastructure and business services for enabling data quality. All Siebel CRM data quality users must enable data quality at the user level using Siebel Data Quality (SDQ). This chapter contains the following topic:

- *Siebel Data Quality Objects*

## Siebel Data Quality Objects

Integration between Siebel CRM and a third-party data quality vendor is not possible without Siebel Data Quality. All of the following types of objects in the DeDuplication project are licensed with Siebel Data Quality and required for its use:

- Business Services
- Business Components
- Applets
- Classes
- DLLs

# 10 Universal Connector API

## Universal Connector API

This chapter describes the application programming interface (API) functions that third-party software vendors must implement in the dynamic link libraries (DLL) or shared libraries that they provide for use with the Universal Connector. It includes the following topics:

This chapter discusses the Universal Connector API. It includes the following topics:


- *About the Terminology in This Chapter*
- *About the Universal Connector API*
- *About the Universal Connector API*
- *About Universal Connector API Terminology*
- *Vendor Libraries*
- *Connector Initialization and Termination Functions*
- *Session Initialization and Termination Functions*
- *Parameter Setting Functions*
- *External Key Generation Function*
- *Error Message Function*
- *Real-Time Data Matching Functions*
- *Batch Mode Data Matching Functions*
- *Real-Time Data Cleansing Function*
- *Batch Mode Data Cleansing Function*
- *Data Matching and Data Cleansing Algorithms*


## About the Terminology in This Chapter

The following terms are used in this chapter:

- **Driver record.** The record the user just entered in real time or the record for which duplicates have to be found.
- **Candidate records.** The records that potentially match the driver record.
- **Duplicate records.** The subset of candidate records that actually match the driver record after the matching process.
- **Primary record.** The record for which data matching was performed.

# About the Universal Connector API

The following topics describe the application programming interface (API) functions that third-party software vendors must implement in the dynamic link libraries (DLL) or shared libraries that they provide for use with the Universal Connector. This topic includes the following information:


- *About Universal Connector API Terminology*
- *Vendor Libraries*
- *Connector Initialization and Termination Functions*
- *Session Initialization and Termination Functions*
- *Parameter Setting Functions*
- *Error Message Function*
- *Real-Time Data Matching Functions*
- *Batch Mode Data Matching Functions*
- *Real-Time Data Cleansing Function*
- *Batch Mode Data Cleansing Function*
- *Data Matching and Data Cleansing Algorithms*


# About Universal Connector API Terminology

The following terms are used in descriptions of the Universal Connector API:

- **Driver record.** The record the user just entered in real time or the record for which duplicates have to be found.
- **Candidate records.** The records that potentially match the driver record.
- **Duplicate records.** The subset of candidate records that actually match the driver record after the matching process.
- **Primary record.** The record for which data matching was performed.

# Vendor Libraries

Vendors must follow these rules for their DLLs or shared libraries:

- The libraries must be thread-safe. A library can support multiple sessions by using different unique session IDs.
- The libraries must support UTF-16 (UCS2) as the default Unicode encoding.
- If there is a single library for all supported languages, the libraries must be named as follows:
  - *BASE*.dll (on Windows)
  - lib*BASE*.so (on AIX)

where *BASE* is a name chosen by the vendor. If a vendor has many solutions for different types of data, then the vendor can use different base names for different libraries.

- If there are separate libraries for different languages, the library name must include the appropriate language code. For example, for Japanese (JPN), the libraries must be named as follows:
  - *BASE*jpn.dll (on Windows)
  - lib*BASE*jpn.so (on AIX)

The Siebel application loads the libraries from the locations described in *Levels of Enabling and Disabling Data Cleansing and Data Matching*.

The mapping of Siebel application field names to vendor field names is stored as values of the relevant Business Component user properties in the Siebel repository. Storage of these field values is mandatory.

Any other vendor-specific parameter required (for example, port number) for the vendor's library must be stored outside of Siebel CRM.

# Connector Initialization and Termination Functions

This topic describes functions that are called when the vendor library is loaded or when the Siebel Server shuts down:

- *sdq_init_connector Function*
- *sdq_shutdown_connector Function*

## sdq_init_connector Function

This function is called using the absolute installation path of the SDQConnector directory (`./siebsrvr/SDQConnector`) when the vendor library is first loaded to facilitate any initialization tasks. It can be used by the vendor to read any configuration files it might choose to use.

| Item | Description |
|---|---|
| **Syntax** | `int sdq_init_connector (const SSchar * path)` |
| **Parameters** | path: The absolute path of the Siebel Server installation. Vendors can use this path to locate any required parameter file for loading the necessary parameters (like port number and so on). This is a Unicode string because the Siebel Server can be installed for languages other than English. |
| **Return Value** | A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function. |

## sdq_shutdown_connector Function

This function is called when the Siebel Server is shutting down to perform any necessary cleanup tasks.

| Item | Description |
|---|---|
| **Syntax** | `int sdq_shutdown_connector (void)` |
| **Parameters** | This function does not have any parameters. |
| **Return Value** | A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function. |

# Session Initialization and Termination Functions

The Siebel Server works in multi-threaded mode to serve multiple users. To allow for user and invocation-specific parameters, there is the concept of a session context where such values can be stored. The session ID is supplied for all data matching or data cleansing functions. Upon completion of data cleansing or data matching the session is closed.

This topic describes the functions that are used for session initialization and termination:

- *sdq_init_session Function*
- *sdq_close_session Function*

## sdq_init_session Function

This function is called when the current session is initialized. This allows the vendor to initialize the parameters of a session or perform any other initialization tasks required.

| Item | Description |
|---|---|
| **Syntax** | `int sdq_init_session (int * session_id)` |
| **Parameters** | session_id: A unique value provided by the vendor that is used in function calls while the session is active. The value 0 is reserved as an invalid session ID. The Siebel CRM code calls this function with a session ID of 0, so the session ID must be initialized to a nonzero value. |
| **Return Value** | A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function. |

## sdq_close_session Function

This function is called when a particular data cleansing or data matching operation is finished and it is required to close the session. Any necessary cleanup tasks are performed.

| Item | Description |
|---|---|
| **Syntax** | `int sdq_close_session (int * session_id)` |
| **Parameters** | session_id: The session ID obtained by initializing the session. |
| **Return Value** | A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function. |

# Parameter Setting Functions

Most third party software vendors provide lists of parameters to customers so that the customers can configure the vendor library's behavior to suit their business needs.

This topic describes the functions that set parameters at both the global context and at the session context (that is, specific to a session):

- *sdq_set_global_parameter Function*
- *sdq_set_parameter Function*

## sdq_set_global_parameter Function

This function is called to set global parameters. This function call is made after the call to sdq_init_connector.

The vendor must put the configuration file, if using one, in `./siebsrvr/SDQConnectorpath`. When the vendor DLL is loaded, it calls the sdq_init_connector API function (if it is exposed by the vendor) with the absolute path to the SDQConnector directory. It is then up to the vendor to read the appropriate configuration file. The configuration file name is dependent on vendor specifications.

An XML character string is used to specify the parameters. This provides an extensible way of providing parameters with each function call.

Using the sdq_set_global_parameter API, any global parameters specific to the vendor can be put as a user property to DeDuplication business service, where the format of the business service user property is as follows:

```
"Global", "Parameter Name", "Parameter Value"
```

These global parameters are set to the vendor only after the vendor DLL loads. You can define user properties for the DeDuplication business service as follows:

Property: My Connector 1

Value: MyDQMatch

Property: MyDQMatch Parameter 1

Value: "Global", "zGlobalParam1", "zGlobalParam1Val"

| Item | Description |
|---|---|
| **Syntax** | `int sdq_set_global_parameter (const SSchar* parameterList)` |
| **Parameters** | parameterList: An XML character string that contains the list of parameters and values specific to this function call. An example of the XML is as follows:<br><br>`<Data>`<br>` <Parameter>`<br>` <GlobalParam1>GlobalParam1Val</GlobalParam1>`<br>` </Parameter>`<br>`</Data>` |
| **Return Value** | A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function. |

# sdq_set_parameter Function

This function is called, after the call to sdq_init_session, to set parameters that are applicable at the session context.

The vendor must put the configuration file, if using one, in `./siebsrvr/SDQConnectorpath`. When the vendor DLL is loaded, it calls the sdq_init_connector API function (if it is exposed by the vendor) with the absolute path to the SDQConnector directory. It is then up to vendor to read the appropriate configuration file. The configuration file name is dependent on vendor specifications.

Using the sdq_set_parameter API, any session parameters specific to the vendor can be put as a user property to the DeDuplication business service, where the format of the business service user property is as follows:

```
"Session", "Parameter Name", "Parameter Value"
```

These session parameters are set to the vendor, after each session opens with the vendor. You can define user properties for the DeDuplication business service as follows:

Property: My Connector 1 Value: MyDQMatch

Property: MyDQMatch Parameter 2

Value: "Session", "zSessParam2", "zSessParam2Val"

| Item | Description |
|---|---|
| **Syntax** | `int sdq_set_parameter (int session_id, const SSchar* parameterList)` |
| **Parameters** | session_id: The session ID obtained while initializing the session.<br><br>parameterList: An XML character string that contains the list of parameters and values that are specific to this function call. An example of the XML is as follows:<br><br>`<Data>`<br>` <Parameter>`<br>` <Name>RECORD_TYPE</Name>`<br>` <Value>Contact</Value>`<br>` </Parameter>` |

| Item | Description |
|---|---|
| | <Parameter><br><Name>SessionParam1</Name><br><Value>SessionValue1</Value><br></Parameter><br></Data> |
| **Return Value** | A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function. |

# External Key Generation Function

This function allows you to enable third party vendors to generate deduplication keys in Siebel CRM. This topic describes the functions that set parameters for external deduplication key.

To facilitate the matching process, Siebel CRM generates a match key for each record (Account, Contact, and Prospect) using the token expression defined in the Vendor Parameter. This approach has some limitations, foremost is each record can only have one match key; also, the keys have a simple general format that do not take advantage of each vendor's specific matching process. The match keys are stored in Siebel CRM key tables for use in the matching candidate selection step, where Siebel queries the key table for potential matching candidates that have similar key value as the driver record.

Furthermore, Oracle Enterprise Data Quality (OEDQ) generates its own match keys; it can generate multiple match keys per driver record, and each key has an associated key priority number to indicate how significant the key is in the matching process. For this reason, a new optional function is added to the Universal Connector API to allow Siebel CRM to call third party software vendor to get the set of match keys generated by the vendor. Any vendor who can generate the match keys itself, can implement this function to take advantage of this new feature.

The OEDQ connector is further enhanced to improve matching effectiveness, bringing stronger candidates to the top of the list by sorting on the *Key Priority Number* field. Not only does this improve matching accuracy but it also provides better performance.

OEDQ provides richer information about the match result such as the matching rule used by the engine, and the list of attributes that contributed to the match.

## sdq_get_dedup_keys Function

This function allows you to invoke a third party vendor to generate the deduplication keys.

To enable this feature, administrator has to configure new Vendor Parameters:

1. **Administration - Data Quality** > **Third Party Administration** select Vendor: 'EDQ'
2. Add new Vendor Parameter: Key Generation Process by Third Party with the value 'Yes'

| Name | Value |
|---|---|
| Key Generation Process by Third Party | Yes |

3. Additionally, if the vendor also utilizes the Key Priority feature, we need to add a new Vendor Parameter.: 'Support Key Priority' with the value 'Yes'

| Name | Value |
|---|---|
| Support Key Priority | Yes |

**Note:** The Key Priority Number should have a value between 1 and 100, where 1 is the highest priority key (lower key priority value means a higher priority key).

**Note:** To support Multiple Child Entities Deduplication, you must use the data format shown in the following information.

| Item | Description |
|---|---|
| **Syntax** | `int sdq_get_dedup_keys (int session_id,`<br>`SSchar* parameterList,`<br>`SSchar* inputRecordSet,`<br>`SSchar* outputRecordSet)` |
| **Parameters** | session_id: The session ID obtained by initializing the session.<br><br>parameterList: An XML character string contains the list of parameters and values that are specific to this function call. An XML example follows:<br><br>`<Data>`<br>`<Parameter>`<br>`<Name>KeyGenerationParam1</Name>`<br>`<Value>KeyGenerationValue1</Value>`<br>`</Parameter>`<br><br>`<Parameter>`<br>`<Name>KeyGenerationParam2</Name>`<br>`<Value>KeyGenerationValue2</Value>`<br>`</Parameter>`<br>`</Data>`<br><br>**Note:** The parameterList can be set to NULL if all required parameters are already set at the session level.<br><br>inputRecordSet: An XML character string containing the driver record. An XML example follows:<br><br>`<Data>`<br>`<DriverRecord>`<br>`<Account.Id>1-X42</Account.Id>`<br>`<<Name>Siebel</Name>`<br>`<Location>Headquarters</Location>`<br>`<DriverRecord>`<br>`<Data>` |

| Item | Description |
|---|---|
| | **Note:** The Id field can be empty. An example is an incoming record without an assigned ID value.<br><br>Here is an example of the XML in hierarchical data format (to support multiple child entities deduplication):<br><br>`<Data>`<br>`<DriverRecord>`<br>`<Account.Id>12-ABCC</Account.Id>`<br>`<Name>Hibbing Manufacturing</Name>`<br>`<Location>San Jose</Location>`<br>`<ListOfCUTAddress>`<br>`<CUTAddress>`<br>`<City>York1</City>`<br>`<Country>USA</Country>`<br>`<Zip>78201</Zip>`<br>`<State>AL</State>`<br>`<Street> 1200 Davisville Road</Street`<br>`</CUTAddress>`<br>`<CUTAddress>`<br>`<City>York2</City>`<br>`<Country>USA</Country>`<br>`<Zip>78202</Zip>`<br>`<State>AL</State>`<br>`<Street> 1201 Davisville Road</Street`<br>`</CUTAddress>`<br>`</ListOfCUTAddress>`<br>`</DriverRecord>`<br>`</Data>`<br><br>outputRecordSet: An XML character string populated by the vendor that contains the dedup keys generated for the driver record. An XML example follows:<br><br>`<Data>`<br>`<KeyRecord>`<br>`<Account.Id>SAME ID AS DRIVER</Account.Id>`<br>`<KeyValue>Key1</KeyValue>`<br>`<KeyPriority>90</KeyPriority>`<br>`</KeyRecord>`<br>`<KeyRecord>`<br>`<Account.Id> SAME ID AS DRIVER </Account.Id>`<br>`<KeyValue>Key2</KeyValue>`<br>`<KeyPriority>87</KeyPriority>`<br>`</KeyRecord>`<br>`<KeyRecord>`<br>`<Account.Id> SAME ID AS DRIVER </Account.Id>`<br>`<KeyValue>Key3</KeyValue>`<br>`<KeyPriority>80</KeyPriority>`<br>`</KeyRecord>`<br>`</Data>`<br><br>**Note:** Optionally, if vendor supports Key Priority feature, it should return <KeyPriority> field for each key in the outputRecordSet |
| **Return Value** | A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function. |

# Error Message Function

This topic describes the function associated with error messages: sdq_get_error_message.

## sdq_get_error_message Function

This function is called if any of the Universal Connector functions return a code other than 0, which indicates an error. This function performs a message lookup and gets the summary and details for the error that just occurred for display to the user or writing to the log.

| Item | Description |
|---|---|
| **Syntax** | `void sdq_get_error_message (int error_code, SSchar * error_summary, SSchar * error_details)` |
| **Parameters** | error_code: The error code returned from the previous function call.<br>error_summary: A pointer to the error message summary, which is up to 256 characters long.<br>error_details: A pointer to the error message details, which are up to 1024 characters long. |
| **Return Value** | This function does not have a return value. |

# Real-Time Data Matching Functions

This topic describes the different functions that are called for real-time data matching when match candidate acquisition takes place in Siebel CRM and in Oracle Data Quality Matching Server.

- *sdq_dedup_realtime Function* is used when match candidate acquisition takes place in Siebel CRM.
- *sdq_dedup_realtime_nomemory Function* is used when match candidate acquisition takes place in Oracle Data Quality Matching Server.

## sdq_dedup_realtime Function

This function is called to perform real-time data matching when match candidate acquisition takes place in Siebel CRM.

**Note:** To support Multiple Child Entities Deduplication, you must use the data format shown in the following image.

This function sends the data for each record as driver records and their candidate records. The function is called only once; multiple calls to the vendor library are not made even when the set of potential candidate records is huge. As all the candidate records are sent at once, all the duplicates for a given record are returned.

| Item | Description |
|---|---|
| **Syntax** | `int sdq_dedup_realtime (int session_id, SSchar* parameterList, SSchar* inputRecordSet, SSchar* outputRecordSet)` |
| **Parameters** | session_id: The session ID obtained by initializing the session.<br><br>parameterList: An XML character string that contains the list of parameters and values that are specific to this function call. An XML example follows:<br><br>`<Data>`<br>`<Parameter>`<br>`<Name>RealTimeDedupParam1</Name>`<br>`<Value>RealTimeDedupValue1</Value>`<br>`</Parameter>`<br><br>`<Parameter>`<br>`<Name>RealTimeDedupParam2</Name>`<br>`<Value>RealTimeDedupValue2</Value>`<br>`</Parameter>`<br>`</Data>`<br><br>**Note:** The parameterList parameter is set to NULL as all required parameters are already set at the session level.<br><br>inputRecordSet: An XML character string containing the driver record and candidate records. An XML example follows:<br><br>`<Data>`<br>`<DriverRecord>`<br>`<Account.Id>1-X42</Account.Id>`<br>`<Account.Name>Siebel</Account.Name>`<br>`<Account.Location>Headquarters</Account.Location>`<br>`</DriverRecord>`<br><br>`<CandidateRecord>`<br>`<Account.Id>1-Y28</Account.Id>`<br>`<Account.Name>Siebel</Account.Name>`<br>`<Account.Location>Atlanta</Account.Location>`<br>`</CandidateRecord>`<br><br>`<CandidateRecord>`<br>`<Account.Id>1-3-P</Account.Id>`<br>`<Account.Name>Siebel</Account.Name>`<br>`<Account.Location>Rome</Account.Location>`<br>`</CandidateRecord>`<br>`</Data>`<br><br>outputRecordSet: An XML character string populated by the vendor in real time that contains the duplicate records with the scores. An XML example follows:<br><br>`<Data>`<br>`<DuplicateRecord>`<br>`<Account.Id>SAME ID AS DRIVER </Account.Id>`<br>`<DQ.MatchScore></DQ.MatchScore>`<br>`<DQ.RuleName>Given name abbreviated, A010 Address exact</DQ.RuleName>`<br>`<DQ.RuleAttributes>NAME,ADDRESS</DQ.RuleAttributes>`<br>`</DuplicateRecord>`<br><br>`<DuplicateRecord>`<br>`<Account.Id>1-Y28</Account.Id>`<br>`<DQ.MatchScore>92</DQ.MatchScore>` |

| Item | Description |
|---|---|
| | <DQ.RuleName>First name abbreviated, A110 Address</DQ.RuleName><br><DQ.RuleAttributes>NAME,ADDRESS</DQ.RuleAttributes><br></DuplicateRecord><br><br><DuplicateRecord><br><Account.Id>1-3-P</Account.Id><br><DQ.MatchScore>88</DQ.MatchScore><br><DQ.RuleName>Name, Phone Exact</DQ.RuleName><br><DQ.RuleAttributes>NAME, PHONE</DQ.RuleAttributes><br></DuplicateRecord><br></Data> |
| **Return Value** | A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function. |

# sdq_dedup_realtime_nomemory Function

This function is called to perform real-time data matching when match candidate acquisition takes place in Oracle Data Quality Matching Server.

| Item | Description |
|---|---|
| **Syntax** | `int sdq_dedup_realtime_nomemory (int session_id, SSchar* parameterList, SSchar* inputRecordSet, SSchar* outputRecordSet)` |
| **Parameters** | session_id: The session ID obtained by initializing the session.<br><br>parameterList: An XML character string that contains the list of parameters and values that are specific to this function call. An XML example follows:<br><br><Data><br><Parameter><br><Name>RealTimeDedupParam1</Name><br><Value>RealTimeDedupValue1</Value><br></Parameter><br><br><Parameter><br><Name>RealTimeDedupParam2</Name><br><Value>RealTimeDedupValue2</Value><br></Parameter><br></Data><br><br>**Note:** The parameterList parameter is set to NULL as all required parameters are already set at the session level.<br><br>inputRecordSet: An XML character string containing the driver record. An XML example follows:<br><br><Data><br><DriverRecord><br><DUNSNumber>123456789</DUNSNumber><br><Name>Siebel</Name><br><<RowId>1-X40</RowId><br></DriverRecord> |

| Item | Description |
|---|---|
| | </Data><br>outputRecordSet: An XML character string populated by the vendor in real time that contains the duplicate records with the scores. An XML example follows:<br><br><Data><br><DuplicateRecord><br><Account.Id>SAME ID AS DRIVER </Account.Id><br><DQ.MatchScore></DQ.MatchScore><br></DuplicateRecord><br><br><DuplicateRecord><br><Account.Id>1-Y28</Account.Id><br><DQ.MatchScore>92</DQ.MatchScore><br></DuplicateRecord><br><br><DuplicateRecord><br><Account.Id>1-3-P</Account.Id><br><DQ.MatchScore>88</DQ.MatchScore><br></DuplicateRecord><br></Data> |
| **Return Value** | A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function. |

# Batch Mode Data Matching Functions

This topic describes the functions that are called for batch mode data matching:

- *sdq_set_dedup_candidates Function*
- *sdq_start_dedup Function*
- *sdq_get_duplicates Function*

## sdq_set_dedup_candidates Function

This function is called to provide the list of candidate records in batch mode. The number of records sent during each invocation of this function is a customer-configurable deployment-time parameter. However, this is not communicated to the vendor at run time.

**Note:** To support Multiple Child Entities Deduplication, you must use the data format shown in the following table.

| Item | Description |
|---|---|
| **Syntax** | `int sdq_set_dedup_candidates (int session_id, SSchar* parameterList, SSchar* xmlRecordSet)` |
| **Parameters** | session_id: The session ID obtained by initializing the session. |

| Item | Description |
| --- | --- |
| | parameterList: An XML character string that contains the list of parameters and values that are specific to this function call. An example of the XML is as follows:<br><br>`<Data>`<br>`<Parameter>`<br>`<Name>BatchDedupParam1</Name>`<br>`<Value>BatchDedupValue1</Value>`<br>`</Parameter>`<br><br>`<Parameter>`<br>`<Name>BatchDedupParam2</Name>`<br>`<Value>BatchDedupValue2</Value>`<br>`</Parameter>`<br>`</Data>`<br><br>**Note:** The parameterList parameter is set to NULL as all required parameters are already set at the session level.<br><br>xmlRecordSet: When match candidate acquisition takes place in Siebel CRM, the xmlRecordSet parameter is used as follows:<br><br>• For full data matching batch jobs: An XML character string containing a list of candidate records. There is no driver record in the input set. An example of the XML is as follows:<br><br>`<Data>`<br>`<CandidateRecord>`<br>`<Account.Id>2-24-E</Account.Id>`<br>`<Account.Name>Siebel</Account.Name>`<br>`<Account.Location>Somewhere</Account.Location>`<br>`</CandidateRecord>`<br>`<CandidateRecord>`<br>`<Account.Id>1-E-2E</Account.Id>`<br>`<Account.Name>Siebel</Account.Name>`<br>`<Account.Location>Somewhere else</Account.Location>`<br>`</CandidateRecord>`<br>`<CandidateRecord>`<br>`<Account.Id>2-34-F</Account.Id>`<br>`<Account.Name>Siebel</Account.Name>`<br>`<Account.Location>Someplace</Account.Location>`<br>`</CandidateRecord>`<br>`</Data>`<br><br>• For incremental data matching batch jobs: As more candidate records are queried from the Siebel database and sent to the vendor software, the driver records must be marked so that the vendor software knows which records must return duplicate records:<br><br>`<Data>`<br>`<DriverRecord>`<br>`<Account.Id>2-24-E</Account.Id>`<br>`<Account.Name>Siebel</Account.Name>`<br>`<Account.Location>Somewhere</Account.Location>`<br>`</DriverRecord>`<br>`<CandidateRecord>`<br>`<Account.Id>1-E-9E</Account.Id>`<br>`<Account.Name>Siebel</Account.Name>`<br>`<Account.Location>Somewhere else</Account.Location>`<br>`</CandidateRecord>`<br>`<DriverRecord>`<br>`<Account.Id>1-E-2E</Account.Id>`<br>`<Account.Name>Siebel</Account.Name>`<br>`<Account.Location>Somewhere else</Account.Location>` |

| Item | Description |
|---|---|
| | </DriverRecord><br><CandidateRecord><br><Account.Id>1-12-2H</Account.Id><br><Account.Name>Siebel</Account.Name><br><Account.Location>Somewhere else</Account.Location><br></CandidateRecord><br><DriverRecord><br><Account.Id>2-34-F</Account.Id><br><Account.Name>Siebel</Account.Name><br><Account.Location>Someplace</Account.Location><br></DriverRecord><br></Data><br><br>**Note:** The order of the driver records and candidate records is not significant. If a candidate has already been sent, it is not necessary to send it again even though it is a candidates associated with multiple driver records.<br><br>xmlRecordSet: When match candidate acquisition takes place in Oracle Data Quality Matching Server, the xmlRecordSet parameter is used as follows:<br><br>• For full data matching batch jobs, an empty string is sent.<br>• For incremental data matching batch jobs, only driver records are sent.<br>An example of the XML is as follows:<br><br><Data><br><DriverRecord><br><DUNSNumber>123456789</DUNSNumber><br><Name>Siebel</Name><br><RowId>1-X40</RowId><br></DriverRecord><br><DriverRecord><br><DUNSNumber>987654321</DUNSNumber><br><Name>Oracle</Name><br><RowId>1-X50</RowId><br></DriverRecord><br><DriverRecord><br><DUNSNumber>123123123</DUNSNumber><br><Name>IBM</Name><br><RowId>1-X60</RowId><br></DriverRecord><br></Data> |
| **Return Value** | A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function. |

# sdq_start_dedup Function

This function is called to start the data matching process in batch mode, and essentially signals that all the records to be used for data matching have been sent to the vendor's application.

| Item | Description |
|---|---|
| **Syntax** | `int sdq_start_dedup (int session_id)` |
| **Parameters** | session_id: The session ID obtained by initializing the session. |
| **Return Value** | This function does not have a return value. |

# sdq_get_duplicates Function

This function is called to get the primary record with the list of its duplicate records along with their match scores. This is done in batch mode. The number of records received for each call to this function is set in the BATCH_MATCH_MAX_NUM_OF_RECORDS session parameter before the function is called.

| Item | Description |
|---|---|
| **Syntax** | `int sdq_get_duplicates (int session_id, SSchar* xmlResultSet)` |
| **Parameters** | session_id: The session ID obtained by initializing the session.<br><br>xmlRecordSet: An XML character string that the vendor library populates with a primary record and a list of its duplicate records along with their match scores.<br><br>If the number of duplicates is more than the value of the parameter BATCH_MATCH_MAX_NUM_OF_RECORDS, the results can be split across multiple function calls with each function call including the primary record as well. The XML is in the following format:<br><br>`<Data>`<br>` <ParentRecord>`<br>` <DQ.MasterRecordsRowID>2-24-E</DQ.MasterRecordsRowID>`<br>` <DuplicateRecord>`<br>` <Account.Id>2-24-E</Account.Id>`<br>` <DQ.MatchScore>92</DQ.MatchScore>`<br>` </DuplicateRecord>`<br><br>` <DuplicateRecord>`<br>` <Account.Id>2-23-F</Account.Id>`<br>` <DQ.MatchScore>88</DQ.MatchScore>`<br>` </DuplicateRecord>`<br>` </ParentRecord>`<br>`</Data>` |
| **Return Value** | A return value of 0 (zero) indicates successful execution, while a return value of 1 indicates that there are no duplicate records. Any other value is a vendor error code.<br><br>The error message details from the vendor are obtained by calling the sdq_get_error_message function.<br><br>**Note:** Data quality code only processes the returned XML character string while the return value is 0. Even if there are fewer records to return than the value of the BATCH_MATCH_MAX_NUM_OF_RECORDS parameter, the vendor driver sends a return value of 0 and then return a value of 1 in the next call. |

| Item | Description |
|---|---|

# Real-Time Data Cleansing Function

This topic describes the functions that are called for real-time data matching: sdq_datacleanse.

## sdq_datacleanse Function

This function is called to perform real-time data cleansing. The function is called for only one record at a time.

| Item | Description |
|---|---|
| **Syntax** | `int sdq_datacleanse (int session_id, SSchar* parameterList, SSchar* inputRecordSet, SSchar* outputRecordSet)` |
| **Parameters** | parameterList: An XML character string that contains the list of parameters and values that are specific to this function call. An example of the XML is as follows:<br><br>`<Data>`<br>`<Parameter>`<br>`<Name>RealTimeDataCleanseParam1</Name>`<br>`<Value>RealTimeDataCleanseValue1</Value>`<br>`</Parameter>`<br><br>`<Parameter>`<br>`<Name>RealTimeDataCleanseParam2</Name>`<br>`<Value>RealTimeDataCleanseValue2</Value>`<br>`</Parameter>`<br>`</Data>`<br><br>**Note:** This parameter is set to NULL as all required parameters are already set at the session level.<br><br>inputRecordSet: An XML character string containing the driver record. An example of the XML is as follows:<br><br>`<Data>`<br>`<DriverRecord>`<br>`<Contact.FirstName>michael</Contact.FirstName>`<br>`<Contact.LastName>mouse</Contact.LastName>`<br>`</DriverRecord>`<br>`</Data>`<br><br>outputRecordSet: A record set that is populated by the vendor in real time and which contains the cleansed record. An example of the XML is as follows:<br><br>`<Data>`<br>`<CleansedDriverRecord>`<br>`<Contact.FirstName>Michael</Contact.FirstName>`<br>`<Contact.LastName>Mouse</Contact.LastName>`<br>`</CleansedDriverRecord>`<br>`</Data>` |

| Item | Description |
|---|---|
| **Return Value** | A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function. |

# Batch Mode Data Cleansing Function

This topic describes the functions that are called for batch mode data cleansing: sdq_data_cleanse.

## sdq_data_cleanse Function

The same function, is called by data quality code for both real-time and batch data cleansing. For batch data cleansing, the call is made with one record at a time.

# Data Matching and Data Cleansing Algorithms

This topic describes the algorithms used in Siebel CRM code for both real-time and batch data cleansing and data matching:

- *Batch Data Matching Algorithm*
- *Real-Time Data Matching Algorithm*
- *Batch Data Cleansing Algorithm*
- *Real-Time Data Cleansing Algorithm*

## Batch Data Matching Algorithm

The algorithm is as follows:

1. Load the vendor library.
2. Call sdq_init_connector.
3. Call sdq_set_global_parameter.
4. Call sdq_init_session.
5. Call sdq_set_parameter (RECORD_TYPE – Account/Contact/List Mgmt Prospective Contact, BATCH_DATAFLOW_NAME, BATCH_MATCH_MAX_NUM_OF_RECORDS)
6. Query the Siebel database to get the candidate records.

   To get the candidate records, a query against the match key is executed. The match key itself is generated when a record is created, or key fields are updated. Universal Connector supports multiple key generation. For more information about match key generation, see *Match Key Generation*.
7. Call sdq_set_dedup_candidates. This function is called multiple times to send the list of all the candidate records.
8. Call sdq_start_dedup to start the data matching process.

9. Call sdq_getduplicate. This function is called multiple times to get all the primary records and their duplicate records and until the function returns -1 indicating that there are no more records.
10. Call sdq_close_session (int * session_id) while logging out of the current session.
11. Call sdq_close_connector.

# Real-Time Data Matching Algorithm

The algorithm is as follows:

1. Load the vendor library.
2. Call sdq_init_connector.
3. Call sdq_set_global_parameter.
4. Call sdq_init_session.
5. Call sdq_set_parameter (RECORD_TYPE – Account/Contact/List Mgmt Prospective Contact).
6. Call sdq_get_dedup_keys. This fuction calls a third party vendor to generate the dedup keys, then EDQ invokes the function sdq_get_dedup_keys to get the dedup keys for driver record. The keys are then used to query for candidate records to send to then third party vendor in the dedup call which follows.
7. Query the Siebel database to get the Candidate records for the driver record.
8. Call sdq_dedup_realtime.
9. Call sdq_close_session while logging out of current session.
10. Call sdq_close_connector.

# Batch Data Cleansing Algorithm

The algorithm is as follows:

1. Load the vendor library.
2. Call sdq_init_connector.
3. Call sdq_set_global_parameter.
4. Call sdq_init_session.
5. Call sdq_set_parameter (RECORD_TYPE – Account/Business Address/Contact/List Mgmt Prospective Contact, BATCH_DATAFLOW_NAME).
6. Call sdq_get_dedup_keys. This function calls a third party vendor to generate the dedup keys for driver record. The keys are then used to query for candidate records to send to third party vendor in the dedup call which follows.
7. Call sdq_datacleanse. This function is called for each record in the result set of the query. It sends the driver record as XML and the output from the function has the cleansed driver record.
8. After cleansing each record, save the record into the Siebel repository.
9. Call sdq_close_session while logging out of current session.
10. Call sdq_close_connector.

# Real-Time Data Cleansing Algorithm

The algorithm is as follows:

1. Load the vendor library.

2. Call sdq_init_connector.
3. Call sdq_set_global_parameter.
4. Call sdq_init_session.
5. Call sdq_set_parameter (RECORD_TYPE – Account/Business Address/Contact/List Mgmt Prospective Contact).
6. Query the Siebel database to get the Driver Record.
7. Call sdq_datacleanse. This function sends the driver record as XML and the output from the function will have the cleansed driver record.
8. Save the record into the Siebel repository.
9. Call sdq_close_session while logging out of current session.
10. Call sdq_close_connector.

# New Key Generation Algorithm

**Note:** This algorithm is applicable only when dedup keys are generated by a third party vendor

The algorithm is as follows:

1. Call sdq_init_session.
2. Call sdq_set_parameter (RECORD_TYPE) - Account/Contact/Prospect).
3. Call sdq_get_dedup_keys. This function sends the driver record as XML and the output from this call is the list of dedup keys generated by third party vendor. EDQ then stores the keys in its dedup key table.
4. Call sql_close_session.

# 11 Finding and Using Data Quality Information

## Finding and Using Data Quality Information

This chapter discusses where to find information relevant to your use of the data quality products. It includes the following topics:

- *Data Quality Resources*
- *Data Quality Seed Data*

## Data Quality Resources

Data Quality is just one of many server processes available in Siebel CRM. Some of these server processes contribute to data quality functionality and are partly documented in this book. For more information about these server processes, and other information relevant to data quality, see the following information:

- *Technical Documentation on Oracle Technology Network*
- *Information about Data Quality on My Oracle Support*

### Technical Documentation on Oracle Technology Network

The following books are on the *Siebel Bookshelf* and available on Oracle Help Center. Refer to them when using data quality:

- *Using Siebel Tools* for information about how to modify standard Siebel CRM objects and create new objects to meet your organization's business requirements.
- *Siebel Fundamentals* for general information about merging records.
- Siebel CRM Deployment Documentation Suite, including:
  - *Siebel Installation Guide* for details on how to install data quality products
  - *Siebel System Administration Guide* for details on how to administer, maintain, and configure your Siebel Servers
  - *Configuring Siebel Business Applications* for information about configuring Siebel CRM using Siebel Tools
  - *Siebel Developer's Reference* for detailed descriptions of business components, user properties, and so on
  - *Siebel Deployment Planning Guide* to familiarize yourself with the basics of the underlying Siebel application architecture.
  - *Siebel System Monitoring and Diagnostics Guide* .
  - *Siebel Security Guide* for information about built-in seed data in the enterprise database, such as employee, position, and organization records.
  - *Siebel Performance Tuning Guide* for information about tuning and monitoring specific areas of the Siebel application architecture and infrastructure, such as the object manager infrastructure.

- *Siebel Data Model Reference* on My Oracle Support (Article ID 546778.1) for information about how data used by the Siebel application is stored in a standard third-party relational DBMS such as DB2, Microsoft SQL Server, or Oracle and some of the data integrity constraints validated by Siebel CRM.
- *Siebel eScript Language Reference* for information about writing scripts to extend data quality functionality.
- *Siebel Applications Administration Guide* for general information about administering Siebel CRM.
- *Siebel Database Upgrade Guide* or *Siebel Database Upgrade Guide for DB2 for z/OS* for information about upgrading your installation and about supported upgrade paths.
- The Certifications tab on My Oracle Support for a definitive list of system requirements and supported operating systems for a release, including the following:
    - Information on supported third-party products
    - Lists of product and feature limitations; either unavailable in the release or in certain environments
- *Oracle Customer Hub (UCM) Master Data Management Reference* provides reference information about Oracle Master Data Applications.

## Information about Data Quality on My Oracle Support

The following documentation is on My Oracle Support:

- *Siebel CRM Update Guide and Release Notes* on My Oracle Support . The most current information on known product anomalies and workarounds and any late-breaking information not contained in this book.
- For more important information on various data quality topics, including time-critical information on key product behaviors and issues, see the following:
    - 476548.1 (Article ID) on My Oracle Support. This document was previously published as Siebel FAQ 1593.
    - 476974.1 (Article ID) on My Oracle Support. This document was previously published as Siebel FAQ 1843.
    - 476926.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Alert 611.

# Data Quality Seed Data

Oracle's Siebel CRM includes a sample database that contains example data of various kinds that you can use in demonstrating, evaluating, or experimenting with the Siebel CRM client and Siebel Tools. While you can use the sample database to test real-time data matching, you cannot used it to test batch data matching, because that requires a running Siebel Server. For more information about the sample database, see *Siebel Installation Guide* .

The enterprise database of your default Siebel application contains some seed data, such as employee, position, and organization records. You can use this seed data for training or testing, or as templates for the real data that you enter. For more information on seed data, including descriptions of seed data records, see *Siebel Security Guide* .