



# **Siebel Data Quality Administration Guide**

Version 7.8

June 2006

**ORACLE®**

Copyright © 2005, 2006, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

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

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

**U.S. GOVERNMENT RIGHTS.** Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

# Contents

## **Chapter 1: What's New in This Release**

## **Chapter 2: Overview of Siebel Data Quality**

Data Cleansing and Data Matching	11
Siebel Data Quality Modes of Operation	12
The SDQ Product Modules	12
The SDQ Matching Server	13
The SDQ Universal Connector	13
SDQ Product Module Comparison	14
How SDQ Relates to Other Entities in the Siebel Application	15

## **Chapter 3: Siebel Data Quality Concepts**

Data Cleansing	17
Data Matching	18
Match Key Generation	19
Match Key Generation with the Universal Connector	19
Match Key Generation with the Matching Server	20
Identification of Candidate Records	21
Identification of Candidate Records with the Universal Connector	21
Identification of Candidate Records with the Matching Server	22
Calculation of Match Scores	22
Calculation of Match Scores with the Universal Connector	22
Calculation of Match Scores with the Matching Server	23
Displaying of Duplicates	23
Data Model for the SDQ Matching Server	24
Fuzzy Query	26

## **Chapter 4: Installing and Upgrading Siebel Data Quality**

Installing the SDQ Matching Server	27
The SDQ Matching Server Libraries	28

Upgrading the SDQ Matching Server from Version 7.7	30
Installing the SDQ Universal Connector	30
Installing Third-Party Software for Use with the Universal Connector	31
Installing Third-Party Application Software for Use with the Universal Connector	31
Installing Third-Party Data Cleansing Files for Use with the Universal Connector	31
The SDQ Universal Connector Libraries	31

## **Chapter 5: Enabling and Disabling Data Matching and Data Cleansing**

Levels of Enabling and Disabling Data Cleansing and Data Matching	33
Enabling Siebel Data Quality at the Enterprise Level	35
Specifying Data Quality Settings	37
Disabling Data Matching and Cleansing Without Restarting the Siebel Server	40
Enabling Siebel Data Quality at the Object Manager Level	40
Enabling Siebel Data Quality at the User Level	42
Disabling Data Cleansing for Specific Records	44
Enabling and Disabling Fuzzy Query	44
Identifying Mandatory Fields for Fuzzy Query	45

## **Chapter 6: Using Siebel Data Quality**

Real-Time Data Cleansing and Data Matching	47
Batch Data Cleansing and Data Matching	48
Data Quality Batch Job Parameters	49
Cleansing Data Using Batch Jobs	51
Generating or Refreshing Keys Using Batch Jobs	51
Matching Data Using Batch Jobs	53
Example of Batch Data Matching Using the Matching Server	53
Example of Batch Data Matching Using the Universal Connector	54
Customizing Data Quality Server Component Jobs for Batch Mode	56
Sample SDQ Component Customizations for Batch Mode	56
Merging of Duplicate Records	59
Sequenced Merges	60
Process of Merging Duplicate Records	60
Filtering Duplicate Records	61

Merging Duplicate Records	61
Using Fuzzy Query	62
Example of Enabling and Using Fuzzy Query with Accounts	63
Troubleshooting Siebel Data Quality	64

## **Chapter 7: Configuring Siebel Data Quality**

Siebel Data Quality Configuration Overview	66
Preconfigured SDQ Connector mappings	66
Mapping of Connector Fields to Business Component Fields	67
Mapping Data Matching Connector Fields to Siebel Business Components	68
Data Matching Field Mapping Syntax for the Matching Server	68
Data Matching Field Mapping Syntax for the Universal Connector	69
Example of Adding a Field Mapping for Data Matching with the Universal Connector	70
Mapping Data Cleansing Connector Fields to Siebel Business Component Fields	71
Data Cleansing Field Mapping Syntax	72
Deactivating SDQ Field Mappings	73
Process of Configuring New SDQ Connectors for the Universal Connector	73
Registering New SDQ Connectors	74
Library Base Name User Properties	74
Configuring Business Components for Data Matching and Data Cleansing	76
Example of Configuring a Business Component for Data Matching Using SSA-NAME3	78
Example of Configuring Business Components for Data Matching Using Third-Party Software and Universal Connector	83
Example of Configuring Business Components for Data Cleansing Using Third-Party Software and Universal Connector	86
Configuring the Pop-Up Windows Displayed in Real-Time Data Matching	89
Configuring the Mandatory Fields for Fuzzy Query	89
Match Purpose	90
Configuring Match Purpose	91

## **Chapter 8: Optimizing Siebel Data Quality Performance**

Optimizing Data Cleansing Performance	93
Optimizing Data Matching Performance	93
Optimizing Matching Server Performance	94

- Database Table Considerations 94
- Data Quality Manager Server Tasks 95
- Data Quality Settings 98

## **Appendix A: Invoking Siebel Data Quality from External Callers**

- Invocation of SDQ from External Callers 99
- Scenario for Data Matching Using the Value Match Method 99
- Value Match Method 100
- Scenario for Data Cleansing Using Data Cleansing Business Service Methods 103
- Get Siebel Fields Method 104
- Parse Method 104
- Parse All Method 105

## **Appendix B: Preconfigured Matching Server Properties and Field Mapping Values**

- Preconfigured Matching Server Properties 107
- Preconfigured SSA Field Mappings 108
  - Viewing Preconfigured SSA Field Mappings 108
  - Preconfigured Account Business Component Field Mappings 109
  - Preconfigured Contact Business Component User Properties 110
  - Preconfigured List Mgmt Prospective Contact Business Component Field Mappings 111

## **Appendix C: Universal Connector API**

- Vendor Dynamic Link Libraries 113
  - Terminology 114
- Connector Initialization and Termination Functions 114
  - sdq\_init\_connector Function 114
  - sdq\_shutdown\_connector Function 115
- Session Initialization and Termination Functions 115
  - sdq\_init\_session Function 115
  - sdq\_close\_session Function 116
- Parameter Setting Functions 116
  - sdq\_set\_global\_parameter Function 117
  - sdq\_set\_parameter Function 118
- Error Message Functions 119
  - sdq\_get\_error\_message Function 119

Real-Time Data Matching Functions	119
sdq_dedup_realtime Function	119
Batch Mode Data Matching Functions	122
sdq_set_dedup_candidates Function	122
sdq_start_dedup Function	126
sdq_get_duplicates Function	126
Real-Time Data Cleansing Functions	128
sdq_datacleanse Function	128
Batch Mode Data Cleansing Functions	129
sdq_data_cleanse Function	129
Data Matching and Data Cleansing Algorithms	130
Batch Data Matching Algorithm	130
Real-Time Data Matching Algorithm	130
Batch Data Cleansing Algorithm	131
Real-Time Data Cleansing Algorithm	131

## **Appendix D: Example of Universal Connector Properties and Field Mapping Values**

Preconfigured Connector Properties	133
Viewing Preconfigured Connector Properties	133
Preconfigured Business Service Properties (7.8.2.2)	133
Preconfigured Business Service Properties (7.8.2)	134
Preconfigured Field Mappings	135
Viewing Preconfigured Field Mappings	136
Preconfigured Business Component User Properties (7.8.2.2)	136
Preconfigured Business Component User Properties (7.8.2)	141

## **Appendix E: Finding and Using Siebel Data Quality Information**

Important Siebel Data Quality Resources	145
Technical Documentation	145
SDQ Seed Data	147

## **Index**





# 1

## What's New in This Release

### What's New in Siebel Data Quality Administration Guide, Version 7.8

Table 1 lists changes described in this version of the documentation to support release 7.8 of the software.

Table 1. New Product Features in Siebel Data Quality Administration Guide, Version 7.8

Topic	Description
<p>New Data Quality Settings</p> <p>See <a href="#">"Specifying Data Quality Settings"</a> on page 37.</p>	<p>The following configuration parameters were added in the Data Quality Settings view:</p> <ul style="list-style-type: none"> <li>■ Force User DeDupe - Account</li> <li>■ Force User DeDupe - Contact</li> <li>■ Force User DeDupe - List Mgmt</li> <li>■ Fuzzy Query - Max Returned</li> <li>■ Fuzzy Query Enabled</li> </ul>
<p>Enhanced data matching (deduplication) capability</p> <p>See <a href="#">"Example of Adding a Field Mapping for Data Matching with the Universal Connector"</a> on page 70.</p>	<ul style="list-style-type: none"> <li>■ You can match and deduplicate child records. Before version 7.8, you could only match and deduplicate parent records.</li> <li>■ You can use date of birth or D-U-N-S number as fields used in data matching.</li> </ul>
<p>New fuzzy query capability</p> <p>See <a href="#">"Fuzzy Query"</a> on page 26.</p>	<p>You can use fuzzy query to make searching of records more intuitive and effective.</p> <p><b>NOTE:</b> By default, fuzzy query is disabled.</p>
<p>Enhanced data cleansing functionality</p> <p>See <a href="#">Appendix A, "Invoking Siebel Data Quality from External Callers."</a></p> <p>See <a href="#">"Data Cleansing and Data Matching"</a> on page 11.</p>	<ul style="list-style-type: none"> <li>■ You can invoke the Data Cleansing business service from external callers, such as workflow. Before version 7.8, you could only invoke this business service using C++ coding.</li> <li>■ In Address records, Data Cleansing functionality automatically supplies a State field value if a user enters valid values for Zip Code, City, and Country. Similarly, it automatically supplies a Zip Code value if a user enters valid values for City, State, and Country.</li> </ul>

Table 1. New Product Features in Siebel Data Quality Administration Guide, Version 7.8

Topic	Description
<p>Batch mode enhancement</p> <p>See <a href="#">“Data Quality Batch Job Parameters”</a> on page 49</p> <p>See <a href="#">“Sample Component Customization for Data Matching”</a> on page 57</p>	<p>You can specify Key Type, Match Threshold, and Search Type parameters in data matching batch jobs. These settings override the parameter settings in the User Preferences screen &gt; Data Quality view and the Administrator - Data Quality Settings view.</p>
<p>Language support for data matching</p> <p>See <a href="#">“The SDQ Matching Server Libraries”</a> on page 28</p>	<p>Thai and Arabic language data matching are supported for the Siebel Data Quality Matching Server. Before version 7.8, these languages were not supported.</p>
<p>New upgrade procedures</p> <p>See <a href="#">“Upgrading the SDQ Matching Server from Version 7.7”</a> on page 30</p>	<p>A topic about considerations for upgrading Siebel Data Quality from version 7.7 to version 7.8 has been added.</p>
<p>Applet user property changes</p> <p>See <a href="#">“Example of Configuring Business Components for Data Matching Using Third-Party Software and Universal Connector”</a> on page 83</p>	<p>The User DeDupe View user property is no longer supported from the 7.8.2 release, as the DeDuplication Results view has been replaced by a pop-up window.</p> <p>A new DeDuplication Results Applet user property has been added, which defines the applet to display for real-time data matching.</p>
<p>New preconfigured field mappings for business components.</p> <p>See <a href="#">Appendix B, “Preconfigured Matching Server Properties and Field Mapping Values”</a> and <a href="#">Appendix D, “Example of Universal Connector Properties and Field Mapping Values”</a></p>	<p>The preconfigured mappings for Siebel business component field names to third-party vendor field names, changed for release 7.8.2.2.</p>

This version also contains the following changes:

- A new appendix about the Universal Connector application programming interface (API) has been added, see [Appendix C, “Universal Connector API”](#).
- This document has been extensively reorganized for greater ease of use.
- Terminology change: Siebel eBusiness Applications has changed to Siebel Business Applications.

# 2

## Overview of Siebel Data Quality

This chapter provides an overview of the Siebel Data Quality (SDQ) functionality and products. It includes the following topics:

- [“Data Cleansing and Data Matching” on page 11](#)
- [“Siebel Data Quality Modes of Operation” on page 12](#)
- [“The SDQ Product Modules” on page 12](#)
- [“How SDQ Relates to Other Entities in the Siebel Application” on page 15](#)

### Data Cleansing and Data Matching

The data stored in account, contact, prospect, and business address records in Oracle’s Siebel Business Applications represents your existing and potential customers. Because of the importance of this data, maintaining its quality is essential. To ensure data quality, SDQ provides functionality to clean this data and to remove duplicated data.

In SDQ, *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, SDQ automatically supplies a State field value. Likewise, if a user enters valid values for City, State, and Country, SDQ automatically supplies a Zip Code value
- **Address correction.** SDQ 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.** SDQ converts account, contact, and prospect names to mixed case (initial capitals). Address fields can be converted to mixed case, all lowercase, or all uppercase.
- **Standardization.** SDQ 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* 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 user properties and other system parameters.

In SDQ 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.

# Siebel Data Quality Modes of Operation

In SDQ, data cleansing and data matching operates in both real-time or in batch mode.

In *real-time mode*, data quality functionality is invoked 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 SDQ 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 `svrmgr` command-line utility to submit server component batch jobs. You run these batch jobs at intervals based 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" on page 51](#).
- 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" on page 53](#).

## The SDQ Product Modules

The two SDQ product modules available for performing data quality functions within the Siebel enterprise are:

- **Siebel Data Quality (SDQ) Matching Server.** Provides real-time and batch data matching functionality.
- **Siebel Data Quality (SDQ) 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.

## The SDQ Matching Server

The Matching Server provides an embedded matching engine that identifies potential duplicate data within existing accounts, contacts, and prospects in the Siebel transactional database.

The Matching Server uses embedded SSA-NAME3 software from Identity Systems (formerly Search Software America). The SSA-NAME3 dynamic link libraries (DLL) are embedded in Siebel Business Applications and are installed with Siebel Software Installers for Windows and UNIX. The Matching Server does not require additional third-party software installations to function.

The Matching Server works across the languages and platforms supported by Siebel Business Applications. There are different DLLs containing the matching rules for different countries and languages. The term *population* is used for such a set of matching rules.

For more information about:

- Languages supported, see [“The SDQ Matching Server Libraries” on page 28](#).
- Platforms supported, see *System Requirements and Supported Platforms* on Siebel SupportWeb.
- SSA-NAME3 software, see the SSA-NAME3 documentation included on the *Siebel Business Applications Third-Party Bookshelf*.

## The SDQ Universal Connector

The Universal Connector is a connector to third-party software that allows the Siebel application 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.

For releases up to 7.8.2.2, to use the Universal Connector, you must obtain, license, and install third party software in addition to the Siebel SDQ application software.

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 Siebel Systems' Technical Alliances program. For information about third-party solutions, see the Alliances section at <http://www.siebel.com>. For detailed information about products that are certified for the Universal Connector, see the Partners section at <http://www.siebel.com>.

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.

The Universal Connector works across various languages and platforms, 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 Business Applications.

For more information about:

- Platforms supported, see *System Requirements and Supported Platforms* on Siebel SupportWeb.
- Third-party software, see the relevant documentation included on the *Siebel Business Applications Third-Party Bookshelf*.

## SDQ Product Module Comparison

Table 2 provides a summary and comparison of the capabilities of the two SDQ product modules—the Matching Server and the Universal Connector.

Table 2. Siebel Data Quality Product Module Comparison

Topic	Matching Server	Universal Connector
Provides data matching for account, contact, and prospect data within the Siebel application	X	X
Provides data cleansing for account, contact, prospect, and business address data within the Siebel application		X
Identifies duplicate records stored in accounts, contacts, and prospects data table	X	X
Supports multiple languages and platforms <b>NOTE:</b> Firstlogic software does not support multiple languages.	X	X
Invokes SDQ functionality through standard Siebel business services	X	X
Requires additional third-party software		X
Runs in real-time and batch modes	X	X

## How SDQ Relates to Other Entities in the Siebel Application

The two SDQ product modules—the Matching Server and the Universal Connector—integrate into the overall Siebel application environment, as shown in Figure 1.

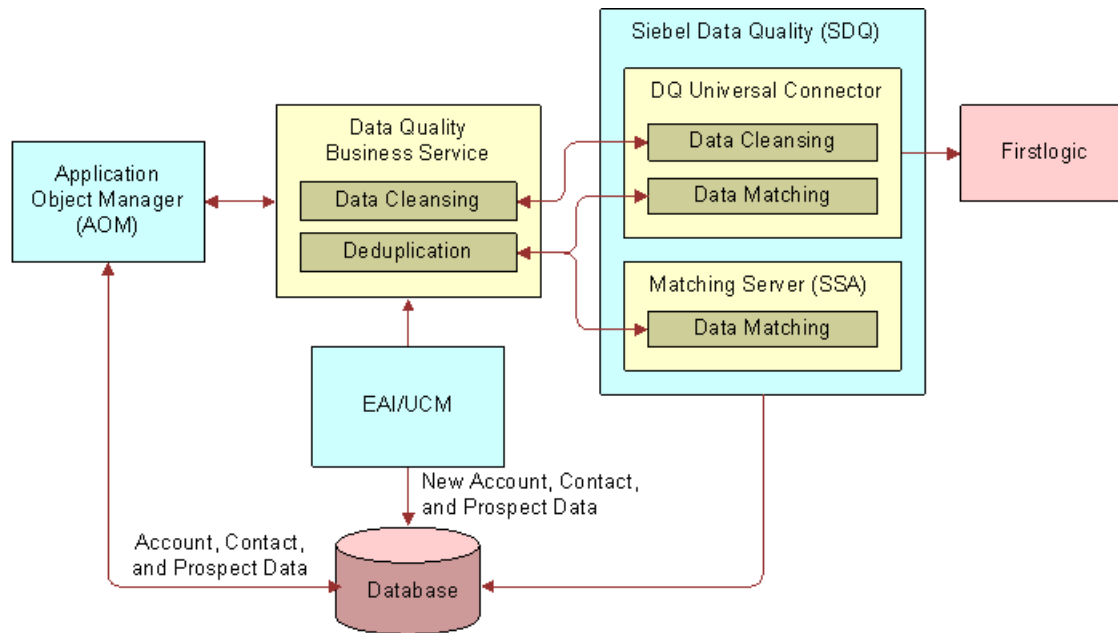


Figure 1. Siebel Data Quality Architecture

In real-time mode, the Matching Server and Universal Connector are invoked by interactive object managers such as the Call Center Object Manager.

In batch mode, the Matching Server and Universal Connector are invoked 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 `svrvmgr` program. For more information, see *Siebel System Administration Guide*.

**NOTE:** You can use both the Matching Server and Universal Connector concurrently in certain configurations. For example, you can simultaneously enable data matching with the Matching Server and the Universal Connector for the same Siebel application object manager.

The Universal Connector and Matching Server obtain 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. In SDQ, 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. In the case of the Matching Server this is an embedded SSA DLL, and in the case of the Universal Connector this is a Firstlogic DLL by default, although the Universal Connector can be associated with other third party software. The cleansing or matching results are returned to the Siebel application.

The match keys used in data matching are generated and stored in the database before matching takes place, and the matching results are also stored in the database. For more information about match keys, see ["Match Key Generation" on page 19](#).

Data matching and data cleansing can also be enabled for the Enterprise Application Integration (EAI) adapter and Siebel Universal Customer Master (UCM) product modules.

For more information about:

- Business services, see *Integration Platform Technologies: Siebel Enterprise Application Integration*.
- Enabling data quality when using EAI and UCM, see the documentation for Enterprise Application Integration and Siebel Universal Customer Master, respectively, on the *Siebel Bookshelf*.



# 3

## Siebel Data Quality Concepts

This chapter provides the conceptual information that you need to use and configure Siebel Data Quality (SDQ). It includes the following topics:

- ["Data Cleansing" on page 17](#)
- ["Data Matching" on page 18](#)
- ["Match Key Generation" on page 19](#)
- ["Identification of Candidate Records" on page 21](#)
- ["Calculation of Match Scores" on page 22](#)
- ["Displaying of Duplicates" on page 23](#)
- ["Data Model for the SDQ Matching Server" on page 24](#)
- ["Fuzzy Query" on page 26](#)

### Data Cleansing

The SDQ Universal Connector supports data cleansing on the Account, Business Address, Contact, and List Mgmt Prospective Contact business components. For each type of record, data cleansing is performed for fields that are specified in user properties for the business component. These user properties specify the mapping between the Siebel field names and the vendor field names. For information about the preconfigured field mappings for Firstlogic, see ["Preconfigured Field Mappings" on page 135](#).

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 Siebel 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.

For information about running data cleansing batch jobs, see [“Cleansing Data Using Batch Jobs” on page 51](#).

## Data Matching

The SDQ Universal Connector and the SDQ Matching Server support data matching on the Account, Contact, and List Mgmt Prospective Contact business components. For each type of record, data matching is performed for fields that are specified in user properties for the business component. These user properties specify the mapping between the Siebel field names and the vendor field names. For information about the preconfigured field mappings for SSA, see [“Preconfigured SSA Field Mappings” on page 108](#), and for Firstlogic, see [“Preconfigured Field Mappings” on page 135](#).

SDQ performs matching for some fields, for example, addresses, that have multi-value group (MVG) associations, however, SDQ does not match on MVGs. Therefore when performing matching for a contact, SDQ 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” on page 53](#).

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 pop-up window for real-time mode, or in the Administration - Data Quality views, from which you can manually merge records into a single record.

The follows topics describe the steps in this sequence in detail.

## Match Key Generation

When data matching is performed in real time or in batch mode, SDQ 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, SDQ 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. These fields are specified in a set of user properties for the appropriate Account, Contact, or Prospect business component. The way in which match keys are calculated differs for the Matching Server and Universal Connector as described in the following topics.

You generate match keys for records in the database by using batch jobs, as described in [“Generating or Refreshing Keys Using Batch Jobs” on page 51](#).

Because key data can become out of sync with the base tables, you need to refresh the key data periodically. A key refresh updates keys only for records that are new or have been modified since your last key generation. Key refresh is much faster than key generation.

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

The batch capability is useful in the following circumstances:

- If you deploy SDQ in a Siebel 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

For instructions about using batch jobs to generate or refresh keys, see [“Generating or Refreshing Keys Using Batch Jobs” on page 51](#).

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 Universal Connector

The SDQ Universal Connector assigns a single key to each account, contact, or prospect record. The key is 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.

Beginning with version 7.8.2.2, the value of the match key depends on a business component-specific Dedup Token Expression, as shown in [Table 3](#).

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.

In earlier versions of the SDQ product, keys for a Universal Connector implementation were stored outside of the Siebel application, in files on the file system. Beginning with version 7.8.2.2, 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)

## Match Key Generation with the Matching Server

For the SDQ Matching Server, multiple keys are generated for each customer record. The number of keys generated depends on the Key Type value in the Data Quality Settings view:

- **Standard.** A more exhaustive range of keys is generated as a wider set of permutations is used. This setting overcomes the most variation in word order, missing words and extra words. This is the default value.
- **Limited.** A subset of keys is generated as only the most common permutations are used. This setting is useful when disk space is limited, but it reduces search reliability.

As an example, if Key Type is set to Standard, the keys generated for the name John Alexander Smith include:

- Smith+John+Alexander
- Smith+Alexander+John
- John+Alexander+Smith
- John+Smith+Alexander
- Alexander+John+Smith

and if Key Type is set to Limited, the keys generated include:

- Smith+John+Alexander
- John+Alexander+Smith
- Alexander+John+Smith

**NOTE:** The keys shown here are only illustrative, as the real keys contain encoded values.

For the various types of record, the match keys are stored in the following tables:

- S\_ORG\_DEDUP\_KEY (for Accounts)
- S\_PER\_DEDUP\_KEY (for Contacts)
- S\_PRSP\_DEDUPKEY (for Prospects)

# Identification of Candidate Records

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

## Identification of Candidate Records with the Universal Connector

SDQ queries the database for candidate records by using a Dedup Query Expression 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 by using Siebel Tools to change the values of these Business Component user properties. 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.

See [Table 3](#) for information about how the default expressions differ for different business components. These values are applicable for Firstlogic.

Table 3. Expressions Used for Keys and Queries (Firstlogic)

Business Component	Dedup Token Expression (Key)	Dedup Query Expression (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 user properties of the DeDuplication business service:

- **Realtime Max Num of Records.** Used in real time, the default value is 200.
- **Batch Max Num of Records.** Used in batch mode, the default is 200.

## Identification of Candidate Records with the Matching Server

For the Matching Server, the value of the Search Type field in the Data Quality Settings view determines how wide a range of keys is searched:

- **Narrow.** A smaller range of keys is searched to provide fastest response.
- **Typical.** A medium range of keys is searched.
- **Exhaustive.** The widest range of keys is searched. In general, if you are using a wider (more exhaustive) key type, also use a wider search type.

## Calculation of Match Scores

After SDQ identifies candidate records, they are sent to the third-party software, which 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; Siebel Systems cannot provide details about how these algorithms work.

The way in which match scores are calculated differs for the Matching Server and Universal Connector as described in the following topics.

## Calculation of Match Scores with the Universal Connector

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 SDQ. 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.

## Calculation of Match Scores with the Matching Server

For the Matching Server, if the match score for a candidate record is greater than or equal to the value of the Match Threshold field in the Data Quality Settings view, the record is flagged as a duplicate of the current record. Match results exceeding the threshold are logged to the S\_DEDUP\_RESULT match results table.

**NOTE:** The rules that control the parsing and weighting criteria that contribute to the match score are precompiled and cannot be modified with the standard SDQ Matching Server module. The custom matching rules must be licensed separately from Search Software America. If your company requires tailored matching rules, contact Siebel Technical Support or Siebel Global Services and ask that they connect you with a representative from Search Software America.

## Displaying of Duplicates

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 pop-up 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 pop-up windows as described in [“Configuring the Pop-Up Windows Displayed in Real-Time Data Matching” on page 89](#).

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” on page 47](#).

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 of Duplicate Records” on page 59](#).

## Data Model for the SDQ Matching Server

This topic provides information about the database tables that the SDQ Matching Server interacts with. [Table 4](#) lists the tables that are relevant to data matching. [Table 5 on page 25](#) lists the database operations that result from the various data matching functions.

Table 4. Tables Used by SDQ Matching Server

Siebel Table Name	Usage by SDQ Matching Server
S_CONTACT	Stores contact records.
S_DEDUP_RESULT	Stores the results of data matching. The following fields are of interest: <ul style="list-style-type: none"> <li>■ <b>DUP_OBJ_ID field.</b> Stores the row ID of a potential duplicate record.</li> <li>■ <b>OBJ_ID field.</b> Stores the row ID of the master record (the record for which matching was performed).</li> <li>■ <b>OBJ_NAME field.</b> Stores the name of the business component for which data matching was performed, for example, Account.</li> <li>■ <b>TOT_SCORE_VAL field.</b> Stores the match score of the potential duplicate record.</li> </ul>
S_ORG_DEDUP_KEY	Stores match keys for account records.
S_ORG_EXT	Stores account records.
S_PER_DEDUP_KEY	Stores match keys for contact records.
S_PRSP_CONTACT	Stores prospective contact records.
S_PRSP_DEDUPKEY	Stores match keys for prospective contact records.



Table 5. Database Operations Resulting from SDQ Matching Functions

Matching Function	Database Operation
Key Generation or Key Refresh	<ul style="list-style-type: none"> <li>■ Data creation in one of the following tables:                             <ul style="list-style-type: none"> <li>■ S_ORG_DEDUP_KEY (Account records)</li> <li>■ S_PER_DEDUP_KEY (Contact records)</li> <li>■ S_PRSP_DEDUPKEY (List Mgmt Prospective Contact records)</li> </ul> </li> <li>■ Data update in one of the following tables:                             <ul style="list-style-type: none"> <li>■ S_CONTACT (Contact records)</li> <li>■ S_ORG_EXT (Account records)</li> <li>■ S_PRSP_CONTACT (List Mgmt Prospective Contact records)</li> </ul> </li> </ul> <p>The DEDUP_KEY_UPD_DT column is updated.</p>
Matching (DeDuplication)	<ul style="list-style-type: none"> <li>■ Data lookup in one of the following tables:                             <ul style="list-style-type: none"> <li>■ S_ORG_DEDUP_KEY (Account records)</li> <li>■ S_PER_DEDUP_KEY (Contact records)</li> <li>■ S_PRSP_DEDUPKEY (List Mgmt Prospective Contact records)</li> </ul> </li> <li>■ Data creation in the S_DEDUP_RESULT table</li> <li>■ Data update in one of the following tables:                             <ul style="list-style-type: none"> <li>■ S_CONTACT (Contact records)</li> <li>■ S_ORG_EXT (Account records)</li> <li>■ S_PRSP_CONTACT (List Mgmt Prospective Contact records)</li> </ul> </li> </ul> <p>The DEDUP_LAST_MTCH_DT column is updated.</p>
Merging	<ul style="list-style-type: none"> <li>■ Data deletion in the S_DEDUP_RESULT table</li> <li>■ Data update and deletion in one of the following tables:                             <ul style="list-style-type: none"> <li>■ S_CONTACT (Contact records)</li> <li>■ S_ORG_EXT (Account records)</li> <li>■ S_PRSP_CONTACT (List Mgmt Prospective Contact records)</li> </ul> </li> </ul>

## 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 SDQ inspects the query for wildcard characters (such as "\*"). If any wildcards are present, SDQ uses standard query functionality for that query, not fuzzy query functionality.
- 3 SDQ 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. SDQ 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" on page 21](#).
- 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 SDQ calls the DeDuplication business service, which works with the third-party data matching server 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 SDQ 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" on page 44](#).

For information about using fuzzy query, see ["Using Fuzzy Query" on page 62](#).

# 4

## Installing and Upgrading Siebel Data Quality

This chapter explains how to install the Siebel Data Quality (SDQ) products. It includes the following topics:

- [“Installing the SDQ Matching Server” on page 27](#)
- [“The SDQ Matching Server Libraries” on page 28](#)
- [“Upgrading the SDQ Matching Server from Version 7.7” on page 30](#)
- [“Installing the SDQ Universal Connector” on page 30](#)
- [“Installing Third-Party Software for Use with the Universal Connector” on page 31](#)
- [“The SDQ Universal Connector Libraries” on page 31](#)

### Installing the SDQ Matching Server

As a preliminary step in installing SDQ software, including the SDQ Matching Server software, you must use the Siebel Image Creator utility and Siebel media files (from your DVD or FTP site) to create a network-based Siebel installation image. For installation instructions, including instructions on creating the installation image, see *Siebel Installation Guide* for the operating system you are using. If you want language-specific versions of the SDQ Matching Server library files installed, you must also add the required language to your installation. For more information about these libraries, see [“The SDQ Matching Server Libraries” on page 28](#).

The InstallShield wizard for Siebel Enterprise Server automatically installs the SDQ Matching Server files on a Siebel Server.

The Siebel Server installation automatically runs the Siebel Software Configuration Utility, which allows you to specify whether you will use SDQ Matching Server or SDQ Universal Connector. If you do not specify SDQ Matching Server during the installation, you can specify it later by selecting Microsoft Windows Start Menu > Programs > Siebel Enterprise Server > Configure Siebel Server to start the utility manually.

Table 6 describes the SDQ Matching Server files and folders that are installed.

Table 6. Siebel Data Quality Matching Server Installation Files

Installation Component	Installation Information
SSA-NAME3 library files	<p>For Windows:  <i>Siebel_Server_root\bin\language_code\n3sgsb.dll</i>                      where <i>language_code</i> is the appropriate language code, such as ENU for U.S. English</p> <p>For Solaris and AIX:  <i>Siebel_Server_root/lib/language_code/n3sgsb.so</i>                      where <i>language_code</i> is the appropriate language code, such as ENU for U.S. English</p> <p>For HP-UX:  <i>Siebel_Server_root/lib/language_code/n3sgsb.sl</i>                      where <i>language_code</i> is the appropriate language code, such as ENU for U.S. English</p> <p>For more information about library files, see <a href="#">“The SDQ Matching Server Libraries” on page 28</a>.</p>

## The SDQ Matching Server Libraries

Character and name patterns differ substantially between languages, therefore the matching rules for the SDQ Matching Server are compiled in a set of dynamic-link libraries (DLLs) adapted for different languages or language families. All of these DLLs have the same name (for example, n3sgsb.dll on Windows), but are installed in language-specific subdirectories as shown in [Table 6](#).

By default, the SDQ Matching Server installation uses a generalized matching library that supports a set of Latin1-based languages (languages predominant in the Americas, Western Europe, Australia, and New Zealand).

In addition, the Siebel installation media includes matching libraries for other languages and code pages. You can retrieve these additional DLLs by installing other language packs on the Siebel Server. [Table 7](#) lists the languages supported.

The SSA Population Codepage user properties of the DeDuplication business service determine which population and code page combination is supported for each language. For example, the SSA Population-Codepage PLK user property has a value of "Poland", "Latin\_2\_1250". For more information about these user properties, see [“Preconfigured Matching Server Properties” on page 107](#).

**NOTE:** The international library intentionally ignores certain words and abbreviations because those words and abbreviations can have a different meaning in other non-Latin1 languages.

For information about code pages, see *Global Deployment Guide*. For information about installation media and installing language packs, see *Siebel Installation Guide* for the operating system you are using.

Table 7. SDQ Matching Libraries and Supported Languages

Matching Library and Code page	Languages Supported and Language Code
International Latin_1_Mixed code page	<ul style="list-style-type: none"> <li>■ DAN - Danish</li> <li>■ DEU - German</li> <li>■ ENU - U.S. English</li> <li>■ ESN - Spanish</li> <li>■ FIN - Finnish</li> <li>■ FRA - French</li> <li>■ ITA - Italian</li> <li>■ NLD - Dutch</li> <li>■ PTB - Brazilian Portuguese</li> <li>■ PTG - Portuguese</li> <li>■ SVE - Swedish</li> </ul>
Eastern European Latin_2_1250 code page	<ul style="list-style-type: none"> <li>■ CSY - Czech</li> <li>■ PLK - Polish</li> </ul>
Arabic	ARA - Arabic
Chinese (Simplified)	CHS - Simplified Chinese
Chinese (Traditional)	CHT - Traditional Chinese
Greek	ELL - Greek
Hebrew	HEB - Hebrew
Japanese	JPN - Japanese
Korean	KOR - Korean
Thai	THA - Thai

**NOTE:** The SDQ Matching Server does not support the ability to find matches across languages that are not supported by the installed library. For example, English and French data can be compared using the international library, but Chinese and Spanish data cannot be compared because Chinese requires a separate library.

## Upgrading the SDQ Matching Server from Version 7.7

As of version 7.8, the SDQ Matching Server provides updated matching algorithms that include newer rules for matching and support the date of birth field as a matching criterion. The base matching libraries have also been updated with new libraries. You must take into consideration the following points when upgrading from version 7.7:

- Version 7.8 uses SSA-NAME3 2.4 libraries
- Key regeneration

Keys generated with an older version of libraries are not compatible with the newer versions. Therefore, to enable data matching you must regenerate keys as part of your upgrade. For more information about regenerating keys as part of an upgrade, see [“Generating or Refreshing Keys Using Batch Jobs” on page 51](#).

Before you regenerate keys, determine whether you need different Data Quality Settings, for example, higher Match Threshold values with the new libraries.

- Match results may vary

The new matching libraries might produce results that are different from the match results from earlier versions. This is due to the enhanced matching routines that are included in version 7.8 of the matching algorithms. It is recommended that you configure the SDQ Matching Server to reestablish the matching baseline, that is, run batch jobs for key generation and data matching against the legacy data.

- Existing results are not affected

If there are unresolved match results stored in the system from previous versions, these results are not affected by the upgrade. However, ongoing deduplication tasks use the newer libraries, so results can vary.

## Installing the SDQ Universal Connector

As a preliminary step in installing SDQ software, including the SDQ Universal Connector software, you must use the Siebel Image Creator utility and Siebel media files (from your DVD or FTP site) to create a network-based Siebel installation image. For installation instructions, including instructions on creating the installation image, see *Siebel Installation Guide* for the operating system you are using.

To use the SDQ Universal Connector, you must install the Data Quality Connector component when running the InstallShield wizard for Siebel Server Enterprise. For information about installing the SDQ Universal Connector on a network, see [“Installing Third-Party Software for Use with the Universal Connector” on page 31](#).

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

This topic explains when and in what location to install third-party software.

### 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 Business Applications. Install the third-party software in the SDQConnector directory where your Siebel Business Applications are 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.

## The SDQ Universal Connector Libraries

The SDQ Universal Connector uses standard Siebel business services for data matching and cleansing. These business services invoke a generalized adapter that can communicate with an external data quality application through a set of dynamic link library (DLL) files.

The names of the DLLs are vendor-specific, but must follow naming conventions as described in [“Vendor Dynamic Link Libraries” on page 113](#).

The Siebel installation process copies these DLL files to a location that depends on the platform you are using, as shown in [Table 8](#).

Table 8. Storage Locations for SDQ Universal Connector DLL Files by Platform

Does Vendor Support Multiple Languages?	DLL Storage Locations (Windows)	DLL Storage Locations (UNIX)
No	For Siebel Server: <i>Siebel_Server_root\bin\</i>  For Developer Web Client: <i>Client_root\bin\</i>	For Siebel Server: <i>Siebel_Server_root/lib</i>
Yes	For Siebel Server: <i>Siebel_Server_root\bin\language_code</i>  For Developer Web Client: <i>Client_root\bin\language_code</i>  where <i>language_code</i> is the appropriate language code, such as ENU for U.S. English.	For Siebel Server: <i>Siebel_Server_root/lib/language_code</i>

**NOTE:** The DLLs for each vendor may be specific to certain platforms 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 machine.

The SDQ Universal Connector requires that you install third-party applications on each Siebel Server machine 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 machine, as well.



# 5

## Enabling and Disabling Data Matching and Data Cleansing

This chapter describes how to enable data cleansing and data matching and describes the data quality settings that you can apply for Siebel Data Quality (SDQ). 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” on page 33](#)
- [“Enabling Siebel Data Quality at the Enterprise Level” on page 35](#)
- [“Specifying Data Quality Settings” on page 37](#)
- [“Enabling Siebel Data Quality at the Object Manager Level” on page 40](#)
- [“Enabling Siebel Data Quality at the User Level” on page 42](#)
- [“Disabling Data Cleansing for Specific Records” on page 44](#)
- [“Enabling and Disabling Fuzzy Query” on page 44](#)
- [“Identifying Mandatory Fields for Fuzzy Query” on page 45](#)

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

In the Siebel application there are various levels at which you can enable or disable data cleansing and data matching as summarized in [Table 9](#). 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.

Table 9. Levels of Enabling and Disabling Data Matching and Cleansing

Screen and View	Setting/Parameter	Value	Parameters Set By
<b>Values set at the enterprise level</b>			
Administration - Server Configuration > Enterprises select Data Quality	Enable button		Application administrator
	Disable button		
Administration - Server Configuration > Enterprises > Parameters	DeDuplication Data Type	<i>Data Matching Vendor Name</i>	
	Data Cleansing Type	<i>Data Cleansing Vendor Name</i>	

**Values set for Data Quality Settings**

**NOTE:** These settings affect all the servers

Table 9. Levels of Enabling and Disabling Data Matching and Cleansing

Screen and View	Setting/Parameter	Value	Parameters Set By
Administration - Data Quality > Data Quality Settings	Enable DataCleansing Enable DeDuplication	Yes/No	Data administrator
<b>Values set at the object manager level</b>			
Administration - Server Configuration > Servers > Select component Data Quality Manager then click > Parameters tab	Data Cleansing Enable Flag Data Cleansing Type DedDuplication Enable Flag DeDuplication Data Type	True/False <i>Data Cleansing Vendor Name</i> True/False <i>Data Matching Vendor Name</i>	Data administrator
Administration - Server Configuration > Servers > Select object manager of application, for example, Sales Object Manager (ENU) then click Parameters tab	Data Cleansing Enable Flag Data Cleansing Type DedDuplication Enable Flag DeDuplication Data Type	True/False <i>Data Cleansing Vendor Name</i> True/False <i>Data Matching Vendor Name</i>	
<b>Values set at the user level</b>			
Tools > User Preferences > Data Quality	Enable DataCleansing Enable DeDuplication	Yes/No	Data steward and end users  <b>NOTE:</b> 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 fields that are defined in DataCleansing and DeDuplication user properties for the relevant business components.

For more information, see the following topics:

[“Enabling Siebel Data Quality at the Enterprise Level” on page 35](#)

[“Specifying Data Quality Settings” on page 37](#)

[“Enabling Siebel Data Quality at the Object Manager Level” on page 40](#)

[“Enabling Siebel Data Quality at the User Level” on page 42](#)

## Enabling Siebel Data Quality at the Enterprise Level

Before performing any batch data matching or data 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 (srmgr) 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 SDQ at the enterprise level*

- 1 Log in to the Siebel application with administrator responsibilities.
- 2 Navigate to the Administration - Server Configuration screen > Enterprises view.
- 3 Click the Component Groups view tab.
- 4 In the Component Groups list, select Data Quality, and then click the Enable button.  
SDQ is now enabled at the enterprise level for data matching and data cleansing.
- 5 Restart the Siebel Server.

*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 > 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.
- **SSA**. Indicates that the Matching Server is used for data matching. This value is set when you choose Siebel Data Quality Matching when you install the Siebel Server.
- **Firstlogic**. Indicates that Firstlogic is used for data cleansing or data matching.
- **Vendor1**. Indicates that third-party software is used for data cleansing or data matching. This value is set when you choose Data Quality Connector when you install the Siebel Server.

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 > Data Quality Settings view.
- 2 In the Value field for each parameter, apply the appropriate settings.

The parameters applicable to all SDQ product modules are described in [Table 10 on page 38](#). The parameters applicable only to the Matching Server are described in [Table 11 on page 39](#)

- 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.

Table 10. Data Quality Settings Applicable to any SDQ Product Module

Parameter	Description
Enable DataCleansing	<p>Determines whether real-time data cleansing is enabled for the Siebel Server the administrator is currently logged into. The default value is Yes.</p> <p>Other values you set for SDQ can override this setting. For more information about this, see <a href="#">“Levels of Enabling and Disabling Data Cleansing and Data Matching” on page 33.</a></p> <p><b>NOTE:</b> Before the 7.8.2.2 release, this parameter was called <b>Disable DataCleansing.</b></p>
Enable DeDuplication	<p>Determines whether real-time data matching is enabled for the Siebel Server the administrator is currently logged into. The default value is Yes.</p> <p>Other values you set for SDQ can override this setting. For more information about this, see <a href="#">“Levels of Enabling and Disabling Data Cleansing and Data Matching” on page 33.</a></p> <p><b>NOTE:</b> Before the 7.8.2.2 release, this parameter was called <b>Disable DeDuplication.</b></p>
Force User Dedupe - Account	<p>Determines whether duplicate records are displayed in a pop-up window when a user saves a new account record. The user can then merge duplicates.</p> <p>If the value is set to No, duplicates are not displayed in a pop-up window, but the user can merge duplicates in the Duplicate Accounts view.</p> <p>The default value is Yes.</p>
Force User DeDupe - Contact	<p>Determines whether duplicate records are displayed in a pop-up window when a user saves a new contact record. The user can then merge duplicates.</p> <p>If the value is set to No, duplicates are not displayed in a pop-up window, but the user can merge duplicates in the Duplicate Contacts view.</p> <p>The default value is Yes.</p>
Force User DeDupe - List Mgmt	<p>Determines whether duplicate records are displayed in a pop-up window when a user saves a new prospect record. The user can then merge duplicates.</p> <p>If the value is set to No, duplicates are not displayed in a pop-up window, but the user can merge duplicates in the Duplicate Prospects view.</p> <p>The default value is Yes.</p>

Table 10. Data Quality Settings Applicable to any SDQ Product Module

Parameter	Description
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 <a href="#">“Enabling and Disabling Fuzzy Query” on page 44</a>
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 <a href="#">“Enabling and Disabling Fuzzy Query” on page 44</a> .

Table 11. Data Quality Settings for the SDQ Matching Server Only

Parameter	Description
Key Type	Determines the number of match keys generated. Possible values are: <ul style="list-style-type: none"> <li>■ <b>Limited.</b> The key generation algorithm performs only the most common permutations and generates fewer keys.</li> <li>■ <b>Standard.</b> The key generation algorithm performs a wider set of permutations to provide the most exhaustive range of keys.</li> </ul> Match keys are generated by applying an algorithm to the matching fields to derive a set of keys that can be compared for similarity by the Matching Server. The Matching Server generates multiple keys for each existing customer record.
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 = 100).  Possible values are: 50-100.  <b>NOTE:</b> If you change the Match Threshold value, you must stop and restart the Siebel Server System Service for the change to take effect. For more information about stopping and starting the Siebel Server System Service, see <i>Siebel System Administration Guide</i> .
Search Type	Indicates whether the match algorithm uses a narrow set of matching rules or a more exhaustive set of rules. A more exhaustive set of rules looks for additional data permutations, but typically takes more time to process.  Possible values are: Exhaustive, Narrow, and Typical.

## 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 Siebel Data Quality at the Object Manager Level

In real-time mode, SDQ is invoked 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 `svrmgr` 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. These procedures require that SDQ is already enabled at the enterprise level. For information about enabling SDQ at the enterprise level, see [“Enabling Siebel Data Quality at the Enterprise Level” on page 35](#).

### ***To enable SDQ 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 > 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 in the table below to enable or disable data matching or cleansing.

Parameter	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 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 `svrmgr` or by completing the following sub-steps:
  - a From the application-level menu, choose `Navigate > Site Map > Administration - Server Management > Servers`.
  - 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*.

### *To enable SDQ 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 administrator responsibility to start or run Siebel Server tasks using the Siebel Server Manager command-line interface.

- 2 Execute a command like one of 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 SSA software:  
change parameter DedDupTypeEnable=True, DeDupTypeType=SSA for component SCCObjMgr\_enu
- To enable data matching if you are using Universal Connector third-party software:  
change parameter DedDupTypeEnable=True, DeDupTypeType=FirstLogic for component SCCObjMgr\_enu
- To enable data cleansing if you are using Universal Connector third-party software:  
change parameter DataCleansingEnable=True, DataCleansingType=FirstLogic for component SCCObjMgr\_enu

To disable data matching or data cleansing, executes commands like these examples with the DeDupTypeEnable or DataCleansingEnable parameters set to False.

For more information on using the command-line interface, see *Siebel System Administration Guide*.

## Enabling Siebel 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 Preferences screen > Data Quality view displays many of the same options that are set in the Administration - Data Quality Settings view. 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 preference data quality settings.

**To specify User Preferences data quality settings**

- 1 Log in to the SDQ application as the user.
- 2 Navigate to the User Preferences screen > Data Quality view.
- 3 In the Data Quality form, set the parameters for that user.

The following table describes the fields.

Field	Comments
Enable Data Cleansing	<p>Select Yes to enable data cleansing for the current user. Otherwise, select No to disable data cleansing.</p> <p>You cannot enable data cleansing here unless data cleansing is enabled in the Data Quality Settings.</p> <p><b>NOTE:</b> Before the 7.8.2.2 release, this parameter was called <b>Disable Data Cleansing</b>.</p>
Enable DeDuplication	<p>Select Yes to enable data matching for the current user. Otherwise select No to disable data matching.</p> <p>You cannot enable data matching here unless data cleansing is enabled in the Data Quality Settings.</p> <p><b>NOTE:</b> Before the 7.8.2.2 release, this parameter was called <b>Disable DeDuplication</b>.</p>
Fuzzy Query Enabled	<p>Select Yes to use a fuzzy query for the current user. Fuzzy query only works if certain conditions are met; see <a href="#">“Enabling and Disabling Fuzzy Query” on page 44</a>.</p> <p>Select No to disable fuzzy queries for the current user.</p>
Fuzzy Query - Max Matches Returned	<p>Specify the maximum number of query result records you want SDQ to return to you. Valid values are 10 to 500.</p> <p>The default value is 100.</p>
Key Type	<p>Applicable only for the SDQ Matching Server.</p> <p>Select Limited to generate a smaller number of keys, or select Standard to generate a larger number of keys.</p> <p>For more information about Key Type settings, see <a href="#">Table 11 on page 39</a>.</p>

Field	Comments
Match Threshold	Applicable only for the SDQ Matching Server.  Select a threshold above which any record with a match score is considered a match. Higher scores indicate closer matches (a perfect match = 100). Possible values are: 50-100.
Search Type	Applicable only for the SDQ Matching Server.  Select Exhaustive, Narrow, or Typical.  This setting indicates whether the match algorithm uses a narrow set of matching rules or a more exhaustive set of rules.

Log out of the application and log back in as the user to initialize the new settings.

**NOTE:** You do not need to 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

Siebel 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” on page 26](#).

When all of the following conditions are satisfied, your Siebel application uses fuzzy query mode automatically, regardless of which SDQ product module 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” on page 37](#).
- Data matching must not be disabled for the current user in the User Preferences - Data Quality view; see [“Enabling Siebel Data Quality at the User Level” on page 42](#).

- 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” on page 45](#).
- The query must leave optional fields blank.

The following procedures describe how to enable and disable fuzzy query in the Data Quality Settings.

### *To enable fuzzy query*

- 1 Navigate to the Administration - Data Quality screen > 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.

### *To disable fuzzy query*

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

For more information about fuzzy query, see [“Using Fuzzy Query” on page 62](#), and [“Example of Enabling and Using Fuzzy Query with Accounts” on page 63](#).

## Identifying Mandatory Fields for Fuzzy Query

You may want to provide users with information about mandatory fields (query fields that must include values for the Siebel application to use fuzzy query mode). [Table 12](#) shows the preconfigured mandatory fields that Siebel Systems provides.

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

Table 12. Mandatory Fields for Fuzzy Query, by Business Component

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 Start Siebel 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 may need to 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.

# 6

## Using Siebel Data Quality

This chapter explains how to use SDQ to perform your data cleansing and data matching tasks. It includes the following topics:

- [“Real-Time Data Cleansing and Data Matching” on page 47](#)
- [“Batch Data Cleansing and Data Matching” on page 48](#)
- [“Data Quality Batch Job Parameters” on page 49](#)
- [“Cleansing Data Using Batch Jobs” on page 51](#)
- [“Generating or Refreshing Keys Using Batch Jobs” on page 51](#)
- [“Matching Data Using Batch Jobs” on page 53](#)
- [“Customizing Data Quality Server Component Jobs for Batch Mode” on page 56](#)
- [“Sample SDQ Component Customizations for Batch Mode” on page 56](#)
- [“Merging of Duplicate Records” on page 59](#)
- [“Process of Merging Duplicate Records” on page 60](#)
- [“Filtering Duplicate Records” on page 61](#)
- [“Merging Duplicate Records” on page 61](#)
- [“Using Fuzzy Query” on page 62](#)
- [“Troubleshooting Siebel Data Quality” on page 64](#)

### Real-Time Data Cleansing and Data Matching

In real-time mode, SDQ is invoked 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:

- If you think the record is not a duplicate, close the dialog box or click Ignore All.

This action commits the new record to the database.

- If you think the record is a duplicate, select the best-matching record from the dialog box using the Pick button.

This action does not commit the new record to the database. The record that you choose becomes the surviving record. Merging is performed as described in [“Sequenced Merges” on page 60](#).

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 SSA Field Mappings” on page 108](#) and [“Preconfigured Field Mappings” on page 135](#).

## 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 by:

- Using the Siebel Server Manager command-line interface, the `svrvmgr` program.
- Running Data Quality Manager component jobs from the Administration - Server Management > Jobs view in the application.

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 SDQ Matching Server requires generated keys in the key tables first before you can run real-time data matching. The SDQ Universal Connector has a similar requirement, but 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 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 may not trigger logic that activates user interface features. For example, the scripting may not trigger UI features such as pop-up windows that show potential matching records.



## Data Quality Batch Job Parameters

Table 13 shows the parameters used in Data Quality batch jobs. The names of the parameters for both Data Quality Manager component jobs and svrmgr commands are given.

Table 13. Data Quality Batch Job Parameters

Job Parameter/ Server Manager Parameter	Required	Description
Buscomp name bcname	Yes	The name of the business component: Possible values include: <ul style="list-style-type: none"> <li>■ Account</li> <li>■ Contact</li> <li>■ List Mgmt Prospective Contact</li> <li>■ Business Address - applicable to Data Cleansing operations only</li> </ul>
Business Object Name bobjname	Yes	The name of the business object. Possible values include: <ul style="list-style-type: none"> <li>■ Account</li> <li>■ Contact</li> <li>■ List Mgmt</li> <li>■ Business Address - applicable to Data Cleansing operations only</li> </ul>
Operation Type opType	Yes	The type of operation: Possible values are: <ul style="list-style-type: none"> <li>■ Data Cleansing - cleanses data</li> <li>■ Key Generate - generates match keys</li> <li>■ Key Refresh - refreshes match keys</li> <li>■ DeDuplication - performs data matching.</li> </ul>
Object Sorting Clause objsortclause	No	Applicable to Data Matching operations only. Indicates how candidate records are sorted for optimal processing by the data matching software. The default value is "Dedup Token"

Table 13. Data Quality Batch Job Parameters

Job Parameter/ Server Manager Parameter	Required	Description
Object Where Clause  objwhereclause	No	Limits the number of records processed by a data quality task. Typically, you use the account's name or the contact's first name to split up large data quality batch tasks using the first letter of the name.  For example, the following object WHERE clause selects only French account records where the account name begins with A:  [Name] Like 'A*' AND [Country] = 'France'  As another example, the following object WHERE clause selects all records where Name begins with Paris or ends with london:  [Name] Like 'Paris*' or [Name] Like '*london'
Data Quality Setting  DQSetting	No	Specifies data quality settings for data cleansing and data matching jobs. This parameter has three values separated by commas:  <ul style="list-style-type: none"> <li>■ <b>First value.</b> Applicable to SSA only. 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.</li> <li>■ <b>Second value.</b> Applicable to the Universal Connector only. It specifies whether the job is a full or incremental data matching job.</li> <li>■ <b>Third value.</b> Applicable to Firstlogic only. It specifies the name of a dataflow file.</li> </ul> For more information about the use of DQSetting, see <a href="#">"Matching Data Using Batch Jobs" on page 53</a> .
Key Type	No	Specifies a value for the Key Type data quality parameter. This is applicable to SSA only. For more information about this parameter, see <a href="#">Table 11</a> .
Search Type	No	Specifies a value for the Search Type data quality parameter. This is applicable to SSA only. For more information about this parameter, see <a href="#">Table 11</a> .
Threshold	No	Specifies a value for the Threshold data quality parameter. This is applicable to SSA only. For more information about this parameter, see <a href="#">Table 11</a> .

## 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:

```
[Disabl e Cl eansi ng] <> ' 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 `srvmgr>` prompt, enter a command like one of those in the following table to perform data cleansing.

The table shows example commands for each of the relevant business components.

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="","", 'account_datacleanse.xml'"
Business Address	run task for comp DQMgr with bcname= "Business Address", bobjname="Business Address", opType="Data Cleansing", objwhereclause="[field_name] LIKE 'search_string*'", DqSetting="","", 'business_address_datacleanse.xml'"
Contact	run task for comp DQMgr with bcname=Contact, bobjname=Contact, opType="Data Cleansing", objwhereclause LIKE "[field_name]='search_string*'", DqSetting="","", 'contact_datacleanse.xml'"
List Mgmt Prospective Contact	run task for comp DQMgr with bcname= "List Mgmt Prospective Contact", bobjname="List Mgmt", opType="Data Cleansing", objwhereclause LIKE "[field_name]='search_string*'", DqSetting="","", 'prospect_datacleanse.xml'"

## 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. For more information about how SDQ generates, refreshes, and stores keys, see [Chapter 3, "Siebel Data Quality Concepts"](#).

**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	<code>run task for comp DQMgr with bcname=Account, bobjname=Account, opType="Key Generate", objwhereclause="[Updated] &gt; '07/18/2005 16:00:00'"</code>
	Refresh	<code>run task for comp DQMgr with bcname=Account, bobjname=Account, opType="Key Refresh", objwhereclause="[Name] LIKE 'search_string*'"</code>
Contact	Generate	<code>run task for comp DQMgr with bcname=Contact, bobjname=Contact, opType="Key Generate", objwhereclause="[Updated] &gt; '07/01/2005 14:10:00'"</code>
	Refresh	<code>run task for comp DQMgr with bcname=Contact, bobjname=Contact, opType="Key Refresh", objwhereclause="[Last Name] LIKE 'search_string*'"</code>
List Mgmt Prospective Contact	Generate	<code>run task for comp DQMgr with bcname="List Mgmt Prospective Contact", bobjname="List Mgmt", opType="Key Generate", objwhereclause="[Updated] &gt; '07/18/2005 16:00:00'"</code>
	Refresh	<code>run task for comp DQMgr with bcname="List Mgmt Prospective Contact", bobjname="List Mgmt", opType="Key Refresh", objwhereclause="[Last Name] LIKE 'search_string*'"</code>

The examples in the table show slightly different WHERE clauses for key generation and key refresh operations:

- 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:

`obj wherecl ause="[field_name] LIKE '*'`

## Matching Data Using Batch Jobs

Depending on your business requirements, you may want to use batch jobs to perform data matching on some or all of the records in the supported business components. If you need to 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:

```
obj wherecl ause="[field_name] LI KE 'A*' "
obj wherecl ause="[field_name] LI KE 'B*' "
...
obj wherecl ause="[field_name] LI KE 'Z*' "
obj wherecl ause="[field_name] LI KE 'a*' "
...
obj wherecl ause="[field_name] LI KE 'z*' "
```

## Example of Batch Data Matching Using the Matching Server

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”](#) on page 51.
- 2 At the `srvmgr>` prompt, enter commands like those in the following table to perform data matching.

Business Component	Example of Server Manager Command
Account	<code>run task for comp DQMgr with DqSetting=""Delete"", bcname=Account, bobjname=Account, opType=DeDuplication, objwhereclause="[Name] like 'search_string*'"</code>
Contact	<code>run task for comp DQMgr with DqSetting=""Delete"", bcname=Contact, bobjname=Contact, opType=DeDuplication, objwhereclause="[Name] like 'search_string*'"</code>
List Mgmt Prospective Contact	<code>run task for comp DQMgr with DqSetting=""Delete"", bcname="List Mgmt Prospective Contact", bobjname="List Mgmt", opType=DeDuplication, objwhereclause="[Name] like 'search_string*'"</code>

## Example of Batch Data Matching Using the Universal Connector

This topic provides an example of batch data matching using the Universal Connector in which Firstlogic is used as the matching engine.

If you are using Firstlogic software you can run *full data matching jobs* or *incremental data matching jobs* as described in the following topics.

### 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 [Table 14](#).

Full data matching jobs are useful when:

- You want to perform data matching on a whole database table.
- You are setting up the SDQ 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,
obj name=Account, opType=DeDupli cation, obj SortCl ause="Dedup Token",
obj wherecl ause="[Name] LI KE '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 [Table 14](#).

Table 14. DqSetting Parameter Details and Sample Values for Firstlogic

DqSetting Parameter Sequence	Valid Values	Comments
First section	Leave blank	This section is not used by Firstlogic. Specify as two adjacent single quotes.
Second section (Enforce Search Spec on Candidate Records)	<ul style="list-style-type: none"> <li>■ Yes</li> <li>■ No (default)</li> </ul>	<p>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.</p> <ul style="list-style-type: none"> <li>■ Use Yes for full data matching batch jobs.</li> <li>■ Use No for incremental data matching batch jobs.</li> </ul>
Third section	<ul style="list-style-type: none"> <li>■ account_match.xml</li> <li>■ account_incremental_match.xml</li> <li>■ contact_match.xml</li> <li>■ contact_incremental_match.xml</li> <li>■ prospect_match.xml</li> <li>■ prospect_incremental_match.xml</li> </ul>	Specifies the name of the data flow associated with the current batch job. Choose the name to specify based on the business component for which you are doing data matching. If your batch job will examine only records created or changed since the last batch job, use a data flow name that includes "incremental;" otherwise, use the shorter data flow name.

This kind of job is considered an *incremental data matching* job, because data matching was done earlier and does not need 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' , ' account_incremental_match.xml ' ',
bcname=Account, bobj name=Account, opType=DeDupl icati on, obj SortCl ause="Dedup Token",
obj wherecl ause="[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.

## 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 `srvmgr` program.

You use the Administration - Server Configuration views to create customized components based on the Data Quality Manager Server component. You specify Data Quality Manager as the Component Type. Sample customization settings are shown in [Table 15 on page 57](#) through [Table 18 on page 59](#). Do not change the original Data Quality Manager component.

For more information about creating custom component definitions, see *Siebel System Administration Guide*.

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:** As of version 7.8, 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 templates, see *Configuring Siebel Business Applications*.

## Sample SDQ Component Customizations for Batch Mode

This topic provides sample settings you can use for customizing SDQ components.

**NOTE:** It is recommended that you use the same component and alias names shown in [Table 15 on page 57](#) through [Table 18 on page 59](#) to allow easier location of log files.



### Sample Component Customization for Data Matching

Table 15 on page 57 through Table 17 on page 58 provide the recommended custom component definitions for Account, Contact, and Prospect objects for the SDQ Matching Server.

Table 15. Recommended Custom Component Definitions for SDQ Matching Server for Accounts

Component Alias	Component Name	Description	Component Parameter	Value
DQMGrAcctKGen	DQ Account Key Generation	Data quality key generation for accounts	Buscomp Name	Account
			Business Object Name	Account
			Operation Type	Key Generate
DQMGrAcctKRef	DQ Account Key Refresh	Data quality key refresh for accounts	Buscomp Name	Account
			Business Object Name	Account
			Operation Type	Key Refresh
DQMGrAcctDDup	DQ Account Key DeDuplication	Data quality deduplication for accounts	Buscomp Name	Account
			Business Object Name	Account
			Operation Type	DeDuplication
			Key Type	Standard or Limited
			Search Type	Exhaustive, Narrow, or Typical

Table 16. Recommended Custom Component Definitions for SDQ Matching Server for Contacts

Component Alias	Component Name	Description	Component Parameter	Value
DQMGrContKGen	DQ Contact Key Generation	Data quality key generation for contacts	Buscomp Name	Contact
			Business Object Name	Contact
			Operation Type	Key Generate
DQMGrContKRef	DQ Contact Key Refresh	Data quality key refresh for contacts	Buscomp Name	Contact
			Business Object Name	Contact
			Operation Type	Key Refresh

Table 16. Recommended Custom Component Definitions for SDQ Matching Server for Contacts

Component Alias	Component Name	Description	Component Parameter	Value
DQMGrContDDup	DQ Contact Key DeDuplication	Data quality deduplication for contacts	Buscomp Name	Contact
			Business Object Name	Contact
			Operation Type	DeDuplication
			Key Type	Standard or Limited
			Search Type	Exhaustive, Narrow, or Typical

Table 17. Recommended Custom Component Definitions for SDQ Matching Server for Prospects

Component Alias	Component Name	Description	Component Parameter <sup>1</sup>	Value
DQMGrPrspKGen	DQ Prospect Key Generation	Data quality key generation for prospects	Buscomp Name	List Mgmt Prospective Contact
			Business Object Name	List Mgmt
			Operation Type	Key Generate
DQMGrPrspKRef	DQ Prospect Key Refresh	Data quality key refresh for prospects	Buscomp Name	List Mgmt Prospective Contact
			Business Object Name	List Mgmt
			Operation Type	Key Refresh
DQMGrPrspDDup	DQ Prospect Key DeDuplication	Data quality deduplication for prospects	Buscomp Name	List Mgmt
			Business Object Name	List Mgmt
			Operation Type	DeDuplication
			Key Type	Standard or Limited
			Search Type	Exhaustive, Narrow, or Typical

1. For users of Siebel Industry Applications, the CUT Address business component must be used instead of Buscomp Name and Business Object name.

## Sample Component Customization for Data Cleansing

Table 18 provides the recommended custom component definitions for the Account, Contact, Prospects, and Address business objects.

Table 18. Recommended Custom Component Definitions for Data Cleansing

Component Alias	Component Name	Description	Component Parameter <sup>1</sup>	Value
DQMGrAcctDCIns	DQ Account Data Cleansing	Data quality data cleansing for accounts	Buscomp Name	Account
			Business Object Name	Account
			Operation Type	Data Cleansing
DQMGrContDCIns	DQ Contact Data Cleansing	Data quality data cleansing for contacts	Buscomp Name	Contact
			Business Object Name	Contact
			Operation Type	Data Cleansing
DQMGrPrspDCIns	DQ Prospect Data Cleansing	Data quality data cleansing for prospects	Buscomp Name	List Mgmt Prospective Contact
			Business Object Name	List Mgmt
			Operation Type	Data Cleansing
DQMGrAddrDCIns	DQ Address Data Cleansing	Data quality data cleansing for addresses	Buscomp Name	Business Address
			Business Object Name	Business Address
			Operation Type	Data Cleansing

1. For users of Siebel Industry Applications, the CUT Address business component must be used instead of Buscomp Name and Business Object name.

## Merging of 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.

You can merge duplicate records in the following ways:

- **Merge Records option (Edit > Merge Records).** Performs the standard merge functionality available in Siebel applications 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 *Fundamentals* on the *Siebel Bookshelf*.

- **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](#)” on page 60. 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.

**CAUTION:** Merging records is an irreversible operation. You must review all records carefully before using the following procedure and initiating a merge.

## 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 to 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 specific characteristics to be eligible for use in a sequenced merge:

- The field can not 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” on page 61](#)

You create 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” on page 61](#)

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” on page 48](#).

## Filtering Duplicate Records

Use the following procedure to filter 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” on page 62](#).

This task is a step in [“Process of Merging Duplicate Records” on page 60](#).

### *To filter duplicate records*

**1** Navigate to the Administration - Data Quality screen.

**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

Use the following procedure to merge 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.

This task is a step in [“Process of Merging Duplicate Records” on page 60](#).

### *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” on page 60](#).

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” on page 44](#).

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” on page 45](#).

- 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.

#### *To use fuzzy query for accounts*

- 1 Navigate to Accounts > Accounts List.
- 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.

#### *To use fuzzy query for contacts*

- 1 Navigate to Contacts > Contacts List.
- 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.

#### *To use fuzzy query for prospects*

- 1 Navigate to List Management > Prospects.
- 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 may 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.

#### *To enable and use fuzzy query for accounts*

- 1 Perform the steps in [“Enabling and Disabling Fuzzy Query”](#) on page 44
- 2 Perform the steps in [“Enabling Siebel Data Quality at the User Level”](#) on page 42.

**NOTE:** For this example, set the Fuzzy Query - Max Returned data quality setting to 10.

- 3 Navigate to Accounts > Account list.
- 4 Enter your query, and then click Go.

For this example, in the Name field, enter Symphony.

Up to 10 records having Name = 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.

## Troubleshooting Siebel 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 Siebel Data Quality functionality.  
**NOTE:** There are different license keys for the Siebel Data Quality Matching Server and the Siebel Data Quality Universal Connector.
- **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” on page 33](#) and [“Specifying Data Quality Settings” on page 37](#).
- **User Preferences.** Verify that data cleansing or data matching has been enabled for the user.  
For more information, see [“Enabling Siebel Data Quality at the User Level” on page 42](#).
- **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.
- **User Properties.** Verify that the DataCleansing or DeDuplication fields have the correct user property values and that the values are formatted correctly. For example, there must be a space after a comma in user properties that have a compound value.

**TIP:** Check Siebel SupportWeb regularly for updates to troubleshooting and other important information. For more information about SupportWeb, see [“Siebel SupportWeb” on page 146](#).



# 7

## Configuring Siebel Data Quality

This chapter describes the configuration that you can perform for Siebel Data Quality (SDQ). It covers the following topics:

- [“Siebel Data Quality Configuration Overview” on page 66](#)
- [“Preconfigured SDQ Connector mappings” on page 66](#)
- [“Mapping of Connector Fields to Business Component Fields” on page 67](#)
- [“Mapping Data Matching Connector Fields to Siebel Business Components” on page 68](#)
- [“Mapping Data Cleansing Connector Fields to Siebel Business Component Fields” on page 71](#)
- [“Deactivating SDQ Field Mappings” on page 73](#)
- [“Process of Configuring New SDQ Connectors for the Universal Connector” on page 73](#)
- [“Registering New SDQ Connectors” on page 74](#)
- [“Configuring Business Components for Data Matching and Data Cleansing” on page 76](#)
- [“Example of Configuring a Business Component for Data Matching Using SSA-NAME3” on page 78](#)
- [“Example of Configuring Business Components for Data Matching Using Third-Party Software and Universal Connector” on page 83](#)
- [“Example of Configuring Business Components for Data Cleansing Using Third-Party Software and Universal Connector” on page 86](#)
- [“Configuring the Pop-Up Windows Displayed in Real-Time Data Matching” on page 89](#)
- [“Configuring the Mandatory Fields for Fuzzy Query” on page 89](#)
- [“Match Purpose” on page 90](#)
- [“Configuring Match Purpose” on page 91](#)

**NOTE:** You must be familiar with Siebel Tools before performing SDQ configuration tasks. For more information about Siebel Tools, see *Using Siebel Tools* and *Configuring Siebel Business Applications*.

# Siebel Data Quality Configuration Overview

Table 19 summarizes the configuration that you can perform for SDQ.

Table 19. Siebel Data Quality Configuration Options

Configuration	See...
Configure field mappings for business components. You can change or add field mappings.	<a href="#">“Mapping Data Matching Connector Fields to Siebel Business Components” on page 68</a> and <a href="#">“Mapping Data Cleansing Connector Fields to Siebel Business Component Fields” on page 71</a>
Configure new connectors for data matching and data cleansing for the Universal Connector.	<a href="#">“Process of Configuring New SDQ Connectors for the Universal Connector” on page 73</a>
Configure business components to support data matching and data cleansing. You can configure existing business components or new business components to support data matching and data cleansing.	<a href="#">“Configuring Business Components for Data Matching and Data Cleansing” on page 76</a>
Configure SSA match purpose.	<a href="#">“Match Purpose” on page 90</a> and <a href="#">“Configuring Match Purpose” on page 91</a>
Configure the mandatory fields for fuzzy search.	<a href="#">“Configuring the Mandatory Fields for Fuzzy Query” on page 89</a>
Configure the pop-up windows displayed in real-time data matching.	<a href="#">“Configuring the Pop-Up Windows Displayed in Real-Time Data Matching” on page 89</a>

## Preconfigured SDQ Connector mappings

In standard Siebel Business Applications, the following business components are preconfigured to support data cleansing and data matching:

- Account
- Contact
- List Mgmt Prospective Contact

In addition, in any Siebel implementation where data cleansing is enabled, the Business Address business component (or CUT Address business component for Siebel Industry Applications) also supports data cleansing.

Each of these business components has preconfigured user properties that define the correct parameters for communication with a data quality vendor. Table 20 shows the default business component user property settings that specify the connector mappings.

Table 20. Preconfigured SDQ Business Component Connector Mappings

User Property	SDQ Module	Default Value	Comments
DeDuplication Connector - SSA	Matching Server	"SSA"	The connector mapping for the SDQ Matching Server using SSA (Search Software America). This user property is not configurable.
DataCleansing Connector - Firstlogic	Universal Connector	"DataCleanse"	The data cleansing connector mapping for Firstlogic software.
DeDuplication Connector - Firstlogic	Universal Connector	"Dedup"	The data matching connector mapping for Firstlogic software.
<b>Properties deprecated after 7.8.2 release</b>			
DataCleansing Connector - Vendor 1	Universal Connector	"namefirm"	The value is configurable for mapping to various vendors. The default value is set for Firstlogic software.
DeDuplication Connector - Vendor 1	Universal Connector	"match"	The value is configurable for mapping to various vendors. The default value is set for Firstlogic software.

Using Siebel Tools, 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.

## Mapping of Connector Fields to Business Component Fields

For Siebel business components that support data cleansing or data matching, there are user properties whose values specify the fields that are used in data cleansing and data matching. In each of these user properties there is a mapping between the names of the Siebel fields used in data cleansing or data matching and the corresponding vendor field names. There are preconfigured mappings for the following business components for each connector:

- Account
- Business address (data cleansing only)
- Contact
- List Mgmt Prospective Contact

For information about the preconfigured field mappings for SSA, see [“Preconfigured SSA Field Mappings” on page 108](#) and for Firstlogic, see [“Preconfigured Field Mappings” on page 135](#).

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. For example, for Firstlogic, you must modify the corresponding dataflow file. For example, for the Account view for real time, the relevant dataflow file for Firstlogic is called `transactional_account_datacleanse.xml`. Refer to the appropriate third party documentation for information about how to update dataflows.

**NOTE:** You must contact the specific vendor for the list of fields they support for data cleansing and data matching and to understand the effect of changing field mappings.

## Mapping Data Matching Connector Fields to Siebel Business Components

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

This is a step in the following processes:

- [“Process of Configuring New SDQ Connectors for the Universal Connector” on page 73](#)
- [“Configuring Business Components for Data Matching and Data Cleansing” on page 76](#)

### Data Matching Field Mapping Syntax for the Matching Server

For data matching field mappings for the Matching Server, user property names are labeled as:

SSA DeDupl i cati on Fi el d  $n$ , where  $n$  is a sequential integer starting from 1.

Each user property value consists of a pair of strings enclosed in double quotation marks and separated by a comma and a space, such as:

*"external vendor field name", "Siebel business component field name"*

For example, given that the DeDuplication Connector - SSA business component user property is set to "SSA", some of the field mappings for the Account business component are shown in [Table 21](#).

Table 21. Sample Matching Server Data Matching Field Mappings for the Account Business Component

User Property Name	Value
SSA DeDuplication Field 1	"C", "Name"
SSA DeDuplication Field 2	"Country", "Country"
SSA DeDuplication Field 3	"City", "City"

**CAUTION:** Make sure there is a comma and a space between the two field values. Data matching and data cleansing can fail if this syntax is not adhered to.

For a list of the preconfigured field mappings for SSA, see ["Preconfigured SSA Field Mappings"](#) on page 108.

## Data Matching Field Mapping Syntax for the Universal Connector

For data matching field mappings for the Universal Connector, user property names are labeled as:

*Vendor\_Deduplication\_key* DeDuplication Field *n*

where:

- *Vendor\_Deduplication\_key* is the value of the DeDuplication Connector *vendor\_name* user property, for example, the DeDuplication Connector - Firstlogic user property has the value Dedup. The *vendor\_name* value must match the value of the DeDuplication Data Type server parameter.
- *n* is a sequential integer starting from 1.

Each DeDuplication Field user property value consists of a pair of strings enclosed in double quotation marks and separated by a comma and a space:

*"external vendor field name", "Siebel business component field name"*

For example, some of the field mappings for the Account business component are as shown in [Table 22](#).

Table 22. Sample Universal Connector Data Matching Field Mappings for the Account Business Component

User Property Name	Value
DeDuplication Connector Firstlogic	"Dedup"
Dedup DeDuplication Field 1	"Account.Name", "Name"
Dedup DeDuplication Field 2	"Account.Location", "Location"
Dedup DeDuplication Field 3	"Account.Street Address", "Street Address"
<b>Properties deprecated after 7.8.2 release</b>	
match DeDuplication Field 1	"MATCH_FIELD_FIRM", "Name"
match DeDuplication Field 2	"MATCH_FIELD_FIRMLOC", "Location"
match DeDuplication Field 3	"MATCH_FIELD_UNPADDRLINE", "Street Address"

**CAUTION:** Make sure there is a comma and a space between the two field values. Data matching and data cleansing can fail if this syntax is not adhered to.

For a list of the preconfigured field mappings for Firstlogic, see [“Preconfigured Field Mappings” on page 135](#).

### *To map a data matching connector field to a Siebel business component field*

- 1 Start Siebel Tools.
- 2 In the Object Explorer, double-click the Business Component object to expand it, and then select the business component of interest.
- 3 In the Object Explorer, click 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).
- 4 In the Business Component User Properties list, create a new record for each field mapping using the syntax for the connector that you are using.

## **Example of Adding a Field Mapping for Data Matching with the Universal Connector**

In addition to the preconfigured fields that are used in data matching, you can configure your SDQ 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.

The following procedure describes how to add a field mapping.

**To add a field mapping for data matching**

- 1 Start Siebel Tools.
- 2 In the Object Explorer, expand Business Component, and then select the business component of interest:
  - For example, to include a date of birth as a matching criterion, select Contact.
  - For example, to include a D-U-N-S number as a matching criterion, select Account.
- 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).
- 4 In the Business Component User Properties list, create a new record and complete the necessary fields as in the following example for Firstlogic.

User Property Name	Value
Dedup DeDuplication Field <i>n</i> (choose the next available integer for <i>n</i> )	<ul style="list-style-type: none"> <li>■ For date of birth, enter: "Contact. Birth Date", "Birth Date"</li> <li>■ For D-U-N-S number, enter: "Account. DUNS Number", "DUNS Number"</li> </ul>

- 5 (Firstlogic only). Modify the corresponding real-time and batch mode data flows to incorporate the new field so that SDQ considers the new field during data matching comparisons.
 

For example, for data matching that considers birth date, the correct data flows to modify are contact\_match.xml and contact\_incremental\_match.xml.

## Mapping Data Cleansing Connector 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 to Firstlogic applications, 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 business component user properties trigger real-time data cleansing when they are modified.

For more information about the property settings for the preconfigured business components, see [Appendix D, "Example of Universal Connector Properties and Field Mapping Values"](#).

## Data Cleansing Field Mapping Syntax

For data cleansing field mappings, user property names are labeled as:

*Vendor\_DataCleansing\_key name DataCleansing Field n*

where:

- *Vendor\_DataCleansing\_key* is the value of the DataCleansing Connector *vendor\_name* user property, for example, the DataCleansing Connector - Firstlogic user property has the value DataCleansing. The *vendor\_name* value must match the value of the Data Cleansing Type server parameter.
- *n* is a sequential integer starting from 1.

Each DataCleansing Field user property value consists of two field names enclosed in double quotation marks and separated by a comma and a space:

*"Siebel Business Component Field", "Vendor Input Field"*

Beginning with version 7.8.2.2, the field that you specify as the Vendor Input Field is also used as the Vendor Output Field for cleansed data.

**CAUTION:** Make sure there is a comma and a space between the field values. Data cleansing can fail if this syntax is not adhered to.

For releases before 7.8.2.2, each user property value consists of two or three field names enclosed in double quotation marks and separated by a comma and a space:

*"Siebel Business Component Field", "Vendor Input Field"*

or

*"Siebel Business Component Field", "Vendor Input Field", "Vendor Output Field"*

The data cleansing field mappings use two and optionally three values. For some types of data, the connector can use the same field for input and output, while for other types of data the connector must use different fields for input and output. You need to consult with the third-party vendor to learn which field or fields to use. If you do not specify the Vendor output field, the Siebel application does not update the cleansed value to the field.



Examples of the data cleansing field mappings for the Account business component are shown in Table 23.

Table 23. Sample Universal Connector Data Cleansing Field Mappings for the Account Business Component

User Property Name	Value
DataCleansing Connector Firstlogic	"DataCleanse"
DataCleanse DataCleansing Field 1	"Name", "Account.Name"
DataCleanse DataCleansing Field 2	"Location", "Account.Location"
<b>Properties deprecated after 7.8.2 release</b>	
namefirm DataCleansing Field 1	"Name", "drl_ifirmname1", "drl_firm_std1"
namefirm DataCleansing Field 2	"Location", "drl_ifirmloc1", "drl_firm_loc_std1"

**To map a data cleansing connector field to a Siebel business component field**

- 1 Start Siebel Tools.
- 2 In the Object Explorer, double-click the Business Component object to expand it, and then select the business component of interest.
- 3 In the Object Explorer, click 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).
- 4 In the Business Component User Properties list, create a new record for each field mapping using the syntax for the connector that you are using.

## Deactivating SDQ Field Mappings

If you deactivate a SDQ field mapping, shift the remaining fields ahead so that your list of field mappings is sequential. For example, if you deactivate Dedup DeDuplication field 2, you must change field 3 to field 2, field 4 to field 3, and so on. If you do not shift the remaining fields, the Siebel application does not recognize nonconsecutive fields and therefore disregards such fields during data matching and data cleansing.

## Process of Configuring New SDQ Connectors for the Universal Connector

You can define your own connectors for data matching and data cleansing for the Universal Connector by configuring the DeDuplication or Data Cleansing business services.

To configure new SDQ connectors, perform the following tasks:

- 1 [“Registering New SDQ Connectors” on page 74](#)
- 2 [“Configuring Business Components for Data Matching and Data Cleansing” on page 76](#)

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

## Registering New SDQ Connectors

SDQ connector definitions are configured as Business Service user properties. You can specify one external application for data matching and a different application for data cleansing for the Universal Connector. This is done by setting the correct input values for each external application in the Value column of the Business Service user properties for that connector.

This is a step in [“Process of Configuring New SDQ Connectors for the Universal Connector” on page 73](#).

**NOTE:** The Business Service user properties 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 properties 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.

## Library Base Name User Properties

The Deduplication and Data Cleansing business services include a generalized adapter that can communicate to an external data quality application through a set of dynamic-link library (DLL) files.

The Library Base Name business service user properties tell the Siebel application how to load the dynamic-link library (DLL). The names of the DLLs are vendor-specific, but must follow naming conventions as described in [“Vendor Dynamic Link Libraries” on page 113](#).

The Siebel application loads the DLLs from the locations described in [Table 8 on page 32](#).

Perform one of the following procedures to determine the global data matching and data cleansing settings for the SDQ Universal Connector.

### *To register a data quality connector for data matching*

- 1 Log in to Siebel Tools with administrative privileges.
- 2 For each match solution, configure a connector as a business service user property.
  - a In the Object Explorer, expand Business Service, and then select the DeDuplication business service.

- b** In the Object Explorer, select Business Service User Prop.

**TIP:** If the Business Service 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).

- c** In the Business Service User Properties list, add a new record with the following settings:

Property Name	Value
<i>Vendor name</i> Connector <i>n</i> , where <i>n</i> is a number starting from 1	<i>Vendor value</i>  A vendor-specific value
<i>Vendor name</i> must match the value specified in the server parameter DeDuplication Data Type	
<i>Vendor value</i> Attribute - Library Base Name	The name of the vendor DLL

**NOTE:** The DeDuplication business service is preconfigured for SSA and Firstlogic applications. To view the preconfigured settings, see and “Preconfigured Matching Server Properties” on page 107 and “Preconfigured Connector Properties” on page 133 respectively.

**To define a data quality connector for data cleansing**

- 1** Start Siebel Tools.
- 2** For each cleansing solution, configure a connector as a business service user property.
  - a** In the Object Explorer, expand Business Service, and then select the Data Cleansing business service.
  - b** In the Object Explorer, select Business Service User Prop.

**TIP:** If the Business Service 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).

- c In the Business Service User Properties list, add a new record with the following settings:

User Property	Value <sup>1</sup>
<i>Vendor name</i> Connector <i>n</i> , where <i>n</i> is a number starting from 1	<i>Vendor value</i>  A vendor-specific value
<i>Vendor name</i> must match the value specified in the server parameter Data Cleansing Type	value
<i>Vendor value</i> Attribute - Library Base Name	The name of the vendor DLL
<b>Properties deprecated after 7.8.2 release</b>	
Vendor <i>X</i> Connector <i>n</i> where <i>X</i> is a number starting from 2 and <i>n</i> is a number starting from 4	namefirm
Vendor <i>X</i> Connector <i>n</i> where <i>X</i> is a number starting from 2 and <i>n</i> is a number starting from 4	address
address Attribute - Library Base Name	The name of the DLL for address cleansing
namefirm Attribute - Library Base Name	The name of the DLL for name cleansing

1. Configure these properties as required according to the third-party vendor requirements.

**NOTE:** The Data Cleansing business service is preconfigured for SSA and Firstlogic applications. To view the preconfigured settings, see [“Preconfigured Matching Server Properties” on page 107](#) and [“Preconfigured Connector Properties” on page 133](#) respectively.

## Configuring Business Components for Data Matching and Data Cleansing

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

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*.

To configure business components for data matching and data cleansing, you must:

- 1 Associate the business component with a connector.

This includes configuring the following user properties:

User Property	Value
<b>For data matching</b>	
DeDuplication Connector - <i>vendor_name</i>	<i>Vendor_Deduplication_key</i>
<i>Vendor_Deduplication_key</i> Type	<i>Business_component_name</i> (Applicable to Universal Connector only)
Dedup Token Expression	Consult the vendor for the value of this field. (Applicable to Universal Connector only)
Dedup Query Expression	Consult the vendor for the value of this field. (Applicable to Universal Connector only)
<b>For data cleansing</b>	
DataCleansing Connector - <i>vendor_name</i>	<i>Vendor_DataCleansing_key</i>
<i>Vendor_DataCleansing_key</i> Type	<i>Business_component_name</i> (Applicable to Universal Connector only)

- 2 Configure the field mappings.

This includes configuring the following user properties

- **Data matching.** *Vendor\_Deduplication\_key* DeDuplication Field *n*
- **Data cleansing.** *Vendor\_DataCleansing\_key* DataCleansing Field *n*

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

**NOTE:** You must base new business components you create only on the CSSBCBase class to support data cleansing and data matching.

## Example of Configuring a Business Component for Data Matching Using SSA-NAME3

This topic gives one example of configuring a business component for data matching with the SSA Connector. You may use this feature differently, depending on your business model.

### *To configure a business component for data matching with SSA*

- 1 Start Siebel Tools.
- 2 In the Object Explorer, double-click the Business Component object to expand it, and then select the business component of interest.
- 3 In the Object Explorer, click 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).

- 4 In the Business Component User Properties list, create new records with the properties and values provided in the following table.

Property	Value	Comments
Deduplication Connector - SSA	"SSA"	
SSA Match Purpose	Enter one of the following values: ■ Company_Mandatory ■ Company_Optional ■ Contact_Mandatory ■ Contact_Optional  <b>NOTE:</b> By default, the Account business component is set to Company_Optional, and the Contact and List Mgmt Prospective Contact business components are set to Contact_Optional.	If a value is marked mandatory, it implies that the value counts against the total score. Values marked Optional do not count toward the total score. SDQ supports only these four Match Purpose values.  For more information about match purpose, see <a href="#">"Match Purpose" on page 90</a> .

- 5 Configure a DeDuplication Key business component and user properties to specify the business component for your key table.

**NOTE:** Due to the complexity of creating database tables, it is recommended that you contact your database administrator for key table creation.

**a** Create a key table.

A key table is a database table that stores the SSA keys used for matching. You can use one of the following existing key tables as a model:

- ❑ S\_ORG\_DEDUP\_KEY
- ❑ S\_PER\_DEDUP\_KEY
- ❑ S\_PRSP\_DEDUPKEY

For example, the Account business component uses the S\_ORG\_DEDUP\_KEY key table.

**NOTE:** The Matching Server requires a key table for each business component (whereas the Universal Connector does not).

**b** Create a new business component based on the key table you created in [Step a](#).

You can use one of the following existing business components as a model:

- ❑ DeDuplication - SSA Account Key
- ❑ DeDuplication - SSA Contact Key
- ❑ DeDuplication - SSA Prospect Key

For example, the Account business component uses DeDuplication - SSA Account Key.

For more information about how to create business components and define user properties, see *Configuring Siebel Business Applications*.

**6** Create a link and the Algorithm Type field for the business component and the key business component you created in [Step 5 on page 78](#).

**a** Create a link using the following syntax:

*Business Component name/DeDuplication - SSA Business Component name Key*

For example, the link for the Account business component is: Account/DeDuplication - SSA Account Key.

**b** Navigate to the Business Component object type and create a multi-value link for the business component you created in [Step 5 on page 78](#) with the properties and values provided in the following table.

Property	Value
Name	DeDuplication - SSA <i>Business Component name</i> Key
Destination Link	<i>Business Component name/DeDuplication - SSA Business Component name</i> Key
Destination Business Component	DeDuplication - SSA <i>Business Component name</i> Key

For example, the properties and values for the Account business component are:

- ❑ Name = DeDuplication - SSA Account Key

- Destination Link = Account/DeDuplication - SSA Account Key
- Destination Business Component = DeDuplication - SSA Account Key
- c Navigate to the Field object type and create a new field for the business component you created in [Step 5 on page 78](#) with the properties and values provided in the following table.

Property	Value
Name	Algorithm Type
Multivalue Link	DeDuplication - SSA <i>Business Component name</i> Key

For example, the multivalue link for the Account business component is: DeDuplication - SSA Account Key.

For more information about links and multivalue links, see *Configuring Siebel Business Applications*.

- 7 Configure the DeDup Key Modification Date and DeDup Last Match Date fields for your business component:
  - a In the Object Explorer, double-click the Business Component object to expand it, and then select the business component you created in [Step 5 on page 78](#).
  - b In the Object Explorer, click Field.



- c In the Fields list, create two new records with the properties and values provided in the following table.

Property	Value		
	Join	Column	Type
DeDup Key Modification Date	Base table for the business component	DEDUP_KEY_UPD_DT	DTYPE_UTCDATETIME
DeDup Last Match Date	Base table for the business component	DEDUP_LAST_MTCH_DT	DTYPE_UTCDATETIME

For example, the values for the Account business component are listed in the following table:

Property	Value		
	Join	Column	Type
DeDup Key Modification Date	S_ORG_EXT	DEDUP_KEY_UPD_DT	DTYPE_UTCDATETIME
DeDup Last Match Date	S_ORG_EXT	DEDUP_LAST_MTCH_DT	DTYPE_UTCDATETIME

After a record is processed during key generation, the DeDuplication business service updates the following fields to the current date and time:

**DeDup Key Modification Date.** This is useful for future batch generations because you can run a key refresh instead of a more time-consuming key generation.

**DeDup Last Match Date.** This is useful for future batch data matching because you can set an Object WHERE Clause to process records that have not changed since the last match date.

- 8 Create the field mappings between the Siebel fields for which data matching is required and the field names recognized by the vendor.

For more information, see [“Mapping Data Matching Connector Fields to Siebel Business Components” on page 68.](#)

- 9 Create a DeDuplication Results business component based on the S\_DEDUP\_RESULT table with the following field values:

Field	Value
Dup Object Id	DUP_OBJ_ID
Object Id	OBJ_ID
Object Name	OBJ_NAME
Request Id	DEDUP_REQ_ID
Total Score	TOT_SCORE_VAL

The Siebel DeDuplication business service stores the ROW\_ID of the matched pairs in the OBJ\_ID and DEDUP\_OBJ\_ID columns. You can use these columns to join your business component to the primary data table to expose more information of the matched records.

**NOTE:** The Siebel matching process uses the S\_DEDUP\_RESULT table to store the matched pairs with a weighted score. The DeDuplication Results business component is required to insert matched pairs into the S\_DEDUP\_RESULT table as well as display the duplicate records in a DeDuplication Results list applet to users.

- 10 Add the new business component to the business object of the view where you want to enable real-time data matching.

In your primary business component, add a user property called DeDuplication Results BusComp and specify the DeDuplication Results business component that you just configured.

- 11 Configure an applet as your DeDuplication Results List Applet based on the business component you configured in [Step 8 on page 85](#).

This applet is used to display the duplicate records for real-time processing.

**TIP:** It is recommended you make a copy of an existing applet, such as the DeDuplication Results (Account) List Applet, and then make changes to the values (applet title, business component, and list columns). You may want to add join tables and fields to your DeDuplication Results business component and map these fields to your list applet so that you can see the duplicate records rather than their row Ids.

- 12 To trigger real-time matching, perform the following:

- a Modify the applet in which users enter or modify the customer data and base it on the CSSFrameListBase for a list applet or CSSFrameBase for a form applet.
- b Add a user property called DeDuplication Results Applet and specify the DeDuplication Results List Applet you configured in [Step 11](#) in the value column.

- 13 Configure Duplicate views and add them to the Administration - Data Quality screen.

**NOTE:** It is recommended you copy and rename the existing Account Duplicates View and the Account Duplicates Detail View as examples for configuring new views.

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

This configuration is used for sequenced merges. For more information about sequenced merges, see [“Process of Merging Duplicate Records” on page 60](#).

You do not need to map the Merge Sequence Number field to a database column. Instead, set the Calculated attribute to TRUE.

## Example of Configuring Business Components for Data Matching Using Third-Party Software and Universal Connector

This topic gives one example of configuring a business component for data matching with Firstlogic. You may use this feature differently, depending on your business model.

The SDQ data matching functionality, also known as deduplication, is implemented by way of the DeDuplication business service.

The third-party software used as an example in this topic is Firstlogic, however the steps in the procedure are similar for other third-party software.

### *To configure a business component to support data matching for Firstlogic*

- 1 Start Siebel Tools.
- 2 In the Object Explorer, select the Business Component object, and then select the business component of interest.

**NOTE:** Make sure the business component is based on the CSSBCBase class property to support real-time data matching, or make sure that the business component is based on a class whose parent is CSSBCBase. This class includes the specific logic to invoke the DeDuplication business service.

- 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).

- In the Business Component User Properties list, create a new record and complete the necessary fields.

For example, the following table shows the preconfigured data matching connectors for the Account business component:

User Property Name	Value	Description
DeDuplication Connector - Firstlogic	"Dedup"	For data matching with Firstlogic applications.
<b>Properties deprecated after 7.8.2 release</b>		
DeDuplication Connector - Vendor1	"match"	For data matching with Firstlogic applications.

- Configure a user property to assign a Type to your business component.

For example, a user property value for the predefined Account business component is:

User Property Name	Value	Description
Dedup Type	Account	Business component type for Firstlogic connector  This user property is not required, however it does provide a way for the vendor DLL to recognize the type of records the Siebel application passes to it. Some vendors do require this type as a predefined keyword. If it is not required as a keyword, developers can instead make the type a unique word so that the DLL identifies the type change when switching from one type to another in Siebel Business Applications.

- Configure the following user properties:

User Property	Value
Dedup Token Expression	Contact the vendor for the appropriate value of this field.
Dedup Query Expression	Contact the vendor for the appropriate value of this field.

- Create the field mappings between the Siebel fields for which data matching is required and the field names recognized by the vendor.

For more information, see ["Mapping Data Matching Connector Fields to Siebel Business Components" on page 68.](#)

- 8 Create a DeDuplication Results business component based on the S\_DEDUP\_RESULT table with the following field values:

Field	Value
Dup Object Id	DUP_OBJ_ID
Object Id	OBJ_ID
Object Name	OBJ_NAME
Request Id	DEDUP_REQ_ID
Total Score	TOT_SCORE_VAL

The Siebel DeDuplication business service stores the ROW\_ID of the matched pairs in the OBJ\_ID and DEDUP\_OBJ\_ID columns. You can use these columns to join your business component to the primary data table to expose more information of the matched records.

**NOTE:** The Siebel matching process uses the S\_DEDUP\_RESULT table to store the matched pairs with a weighted score. The DeDuplication Results business component is required to insert matched pairs into the S\_DEDUP\_RESULT table as well as display the duplicate records in a DeDuplication Results list applet to users.

- 9 Add the new DeDuplication Results business component to the DeDuplication business object.
- 10 Add the new business component to the business object of the view where you want to enable real-time data matching.

In your primary business component, add a user property called DeDuplication Results BusComp and specify the DeDuplication Results business component that you just configured.

- 11 Configure an applet as your DeDuplication Results List Applet based on the business component you configured in [Step 8 on page 85](#).

This applet is used to display the duplicate records for real-time processing.

**TIP:** It is recommended you make a copy of an existing applet, such as the DeDuplication Results (Account) List Applet, and then make changes to the values (applet title, business component, and list columns). You may want to add join tables and fields to your DeDuplication Results business component and map these fields to your list applet so that you can see the duplicate records rather than their row Ids.

- 12 To trigger real-time matching, perform the following:
  - a Modify the applet in which users enter or modify the customer data and base it on the CSSFrameListBase for a list applet or CSSFrameBase for a form applet.
  - b Add a user property called DeDuplication Results Applet and specify the DeDuplication Results List Applet you configured in [Step 11](#) in the value column.

- 13 Configure Duplicate views and add them to the Administration - Data Quality screen.

**NOTE:** It is recommended you copy and rename the existing Account Duplicates View and the Account Duplicates Detail View as examples for configuring new views.

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

This configuration is used for sequenced merges. For more information about sequenced merges, see [“Process of Merging Duplicate Records” on page 60](#).

**NOTE:** You do not need to map the Merge Sequence Number field to a database column. Instead, set the Calculated attribute to TRUE.

## Example of Configuring Business Components for Data Cleansing Using Third-Party Software and Universal Connector

This topic gives one example of configuring a business component for data cleansing. You may use this feature differently, depending on your business model.

The third-party software used as an example in this topic is Firstlogic, however the steps in the procedure are similar for other third-party software.

### *To configure a business component to support data cleansing*

- 1 Start Siebel Tools.
- 2 In the Object Explorer, expand Business Component, and then select the business component of interest.

**NOTE:** Make sure the business component is based on the CSSBCBase class property to support real-time data matching, or make sure that the business component is based on a class whose parent is CSSBCBase. This class includes the specific logic to invoke the DeDuplication business service.

- 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).

- 4 In the Business Component User Properties list, create a new record and complete the necessary fields.

For example, the following table shows the predefined data cleansing connector for the Account business component:

User Property Name	Value	Description
DataCleansing Connector - Firstlogic	"DataCleanse"	For data cleansing with Firstlogic applications.

User Property Name	Value	Description
<b>Properties deprecated after 7.8.2 release</b>		
DataCleansing Connector - Vendor1	"namefirm"	For data cleansing with Firstlogic applications.

- 5 Configure a user property to assign a Type to your business component.

For example, a user property value for the predefined Account business component is:

User Property Name	Value	Description
DataCleanse Type	Account	Business component type for Firstlogic applications.

- 6 Create the field mappings between the Siebel fields that you want to cleanse and the field names of the external software.

For more information, see ["Mapping of Connector Fields to Business Component Fields" on page 67](#).

- 7 (Optional) If you want to prevent data cleansing on a selected record, perform the following:

- a Add an extension column to the base table and map it to a business component field called Disable DataCleansing.

For example, the fields used in the Business Address business component are:

Field	Description
Field Name	Disable DataCleansing
Column	DISA_CLEANSE_FLG
Predefault value	N
Text Length	1
Type	DTYPE_BOOL

- b** Expose this flag on the applet to allow you to disable data cleansing for certain records from the user interface.

(Optional) Configure a field called Last Clnse Date so that the Data Cleansing business service can mark the last date and time that data cleansing was run on a particular record.

Field	Description
Field Name	Last Clnse Date
Join	S_ORG_EXT
Object Name. Column	OBJ_NAME
Column	DEDUP_DATACLNSD_DT
Type	DTYPE_UTCDATETIME

After a record is cleansed, the Data Cleansing business service attempts to update the Last Clnse Date business component field to the current date and time. This field is useful for future batch data cleansing, because you can use an Object WHERE Clause to cleanse only records that have changed since the last cleanse date. For example, the following values appear in the Account business component:

Object Where Clause: [Last Clnse Date] < [Updated]

- 8** (Optional) Use a DataCleansing Conflict Id Field user property to specify the conflict Id field for a business component.

In most implementations, user keys are defined in the database schema for each table. These user keys make sure that no more than one record has the same set of values in specific fields. For example, the S\_ORG\_EXT table used by the Account business component uses columns NAME, LOC (Location), and BU\_ID (organization id) in the user keys. Before you run data cleansing against your database, you may have similar, but not exactly the same records, in your database.

After these records are cleansed, they can cause user key violations because the cleansed values become exactly the same value. You can use the Conflict Id field to resolve this issue. Add the CONFLICT\_ID system column (given this table column exists in the database schema) to the user keys and then configure a user property called DataCleansing Conflict Id Field in that business component. The following example is for the Account business component:

User Property: DataCleansing Conflict Id Field

Property Value: S\_ORG\_EXT.Conflict Id

If a user key violation occurs when the Siebel application writes the cleansed records to the database, the application tries to update the Conflict Id field to the record's row Id to make the record unique and bypass the user key violation. After the entire database is cleansed, you can perform data matching to catch these records and resolve them.

**CAUTION:** Before modifying user keys, contact Oracle's Siebel Technical Support.



## Configuring the Pop-Up 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 pop-up window.

You can change the name of the pop-up windows that are displayed, and you can specify that a pop-up 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.

### *To change the name of the pop-window displayed*

- 1 Start Siebel Tools.
- 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.

### *To add a deduplication pop-up window for an applet*

- 1 Start Siebel Tools.
- 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 settings like the following:
  - **Name.** DeDuplication Results Applet
  - **Value.** DeDuplication Results (Account) List Applet
- 5 Restart the Siebel Server.

## Configuring the Mandatory Fields for Fuzzy Query

For a business component you can configure the mandatory fields for fuzzy query - query fields that must include values for the Siebel application to use fuzzy query mode. [Table 12 on page 46](#) shows the preconfigured mandatory fields that Siebel Systems 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 Start Siebel 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 may need to 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.

## Match Purpose

The concept of *match purpose* is used by the embedded SSA-NAME3 software of the Matching Server. SSA-NAME3 supports different match purposes so that different fields are used in matching for different types of records.

The Account, Contact, and List Mgmt Contact Prospect business components each have a SSA Match Purpose user property that you can set to one of the following values, which correspond to match purposes:

- **Company\_Mandatory.** Available for Account.
- **Company\_Optional.** Available for Account.
- **Contact\_Mandatory.** Available for Contact and List Mgmt Contact Prospect.
- **Contact\_Optional.** Available for Contact and List Mgmt Contact Prospect.

Each of these values specifies which fields in the record count against the total match score. Fields are defined as optional or mandatory as shown in [Table 24](#).

When a field defined as optional contains a null value in either of the records being compared, it does not contribute to the total match score for the record pair.

When a field defined as mandatory contains a null value in either of the records being compared, the null value is treated the same as a non-null value and does contribute to the overall match score for the record pair.

For more information about match purpose, refer to the SSA-NAME3 documentation.

Table 24. Match Purpose Values for SSA Match Purpose User Property

Match Purpose	Field	Mandatory/Optional
Company_Mandatory	Company	Mandatory
	Address	Mandatory
	Address2	Mandatory
	Zip	Optional
	ID	Optional

Table 24. Match Purpose Values for SSA Match Purpose User Property

Match Purpose	Field	Mandatory/Optional
Company_Optional	Company	Mandatory
	Address	Optional
	Address2	Optional
	Zip	Optional
	ID	Optional
Contact_Mandatory	Person Name	Mandatory
	Company	Mandatory
	Address	Mandatory
	Address2	Mandatory
	Zip	Optional
	ID	Optional
	Email	Optional
	Telephone	Optional
Contact_Optional	Person Name	Mandatory
	Company	Optional
	Address	Optional
	Address2	Optional
	Zip	Optional
	ID	Optional
	Email	Optional
	Telephone	Optional

## Configuring Match Purpose

Use the following process to configure the match purpose for an existing business component.

### *To configure the match purpose for a business component*

- 1 Start Siebel Tools.
- 2 In the Object Explorer, double-click the Business Component object to expand it, and then select the business component of interest.
- 3 In the Object Explorer, click Business Component User Prop.

- 4 In the Business Component User Properties list, select SSA Match Purpose and set the value as required.

# 8

## Optimizing Siebel Data Quality Performance

This chapter provides recommendations for optimizing Siebel Data Quality (SDQ) performance. It includes the following topics:

- [“Optimizing Data Cleansing Performance” on page 93](#)
- [“Optimizing Data Matching Performance” on page 93](#)
- [“Optimizing Matching Server Performance” on page 94](#)

### 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 [Table 25](#). Split the tasks into smaller tasks and run them concurrently.

Table 25. Recommended Data Cleansing Object WHERE Clause Solutions

To Cleanse	Use This in Your Object WHERE Clause
Updated records	[Last Clnse Date] < [Updated]
New records	[Last Clnse Date] IS NULL
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” on page 51](#).

### 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 may 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.

For more information about running batch data matching, see [“Matching Data Using Batch Jobs” on page 53](#).

## Optimizing Matching Server Performance

You can improve performance in the following ways:

- Make sure that database tables associated with data matching are large enough and do not contain unnecessary duplicates.
- Use appropriate batch tasks to optimize performance.
- Use appropriate Data Quality settings to optimize performance.

Recommendations for each of these are described in the following topics.

### Database Table Considerations

The following are recommendations for achieving good performance:

- Make sure there is sufficient space in the database tables used by the Matching Server. Use [Table 26](#) and work with a database administrator to make sure there is sufficient space available for these tables.

Table 26. Table Size Consideration

Table	Sizing Consideration
S_PER_DEDUP_KEY S_ORG_DEDUP_KEY S_PRSP_DEDUPKEY	<p>These tables can include many more records than their corresponding base tables, depending on the key type used during the key generation stage, as follows:</p> <ul style="list-style-type: none"> <li>■ <b>Limited key type.</b> Between two and four times more records</li> <li>■ <b>Standard key type.</b> Up to an estimated six times more records</li> </ul>
S_DEDUP_RESULT	<p>After a full deduplication run, this table can contain five to six times the number of records in the three base tables combined.</p> <ul style="list-style-type: none"> <li>■ If a Typical or Exhaustive search type is used, more records are inserted into the results table.</li> <li>■ If a low match threshold is used, the matching process generates a larger number of records that are inserted into the results table.</li> </ul>

- For the DB2 platform, have your DBA use the REORG, REORGCHK, and RUNSTATS commands to improve performance during database maintenance.

Access to S\_PER\_DEDUP\_KEY, S\_ORG\_DEDUP\_KEY, and S\_PRSP\_DEDUPKEY is on the DEDUP\_KEY column, which is the only column of the table's \_M1 index, therefore REORG uses this index. You must have current statistics for all tables associated with SDQ:

S\_PER\_DEDUP\_KEY, S\_ORG\_DEDUP\_KEY, S\_ORG\_EXT, S\_PRSP\_CONTACT, S\_CONTACT, S\_PRSP\_CONTACT, S\_PARTY, S\_PARTY\_PER, and S\_DEDUP\_RESULT

so that you can use runstats commands to update statistics and improve performance.

- For the DB2 platform, if performance seems degraded, run the following command on all tables associated with SDQ.

```
runstats on table Siebel . Table_Name
```

where *Table\_Name* is the name of the table, for example, S\_PER\_DEDUP\_KEY. If that command returns an error message, use this one instead:

```
runstats on table Siebel . Table_Name with distribution indexes all
```

## Data Quality Manager Server Tasks

The following are recommendations for achieving good performance with batch data matching:

- Run concurrent Data Quality Manager server tasks for data matching.

Use different, mutually-exclusive object WHERE clauses to separate the data matching into smaller batches (not more than 50,000 to 75,000 records at a time). For example, you might run separate tasks for each first letter (or letters) of a contact record's Last Name or Name fields as in the following example:

```
run task for component DQMGr with BObjName="Contact", BCName="Contact",
OpType="Key Generate", ObjWhereClause="[Last Name] Like 'A*'"
```

The object WHERE clauses to process all records are as follows:

```
Obj WhereClause="[Last Name] < 'A' "
Obj WhereClause="[Last Name] Like 'A*'"
Obj WhereClause="[Last Name] Like 'a*'"
Obj WhereClause="[Last Name] Like 'B*'"
Obj WhereClause="[Last Name] Like 'b*'"
...
Obj WhereClause="[Last Name] Like 'Y*'"
Obj WhereClause="[Last Name] Like 'y*'"
Obj WhereClause="[Last Name] Like 'Z*'"
Obj WhereClause="[Last Name] > 'z' "
```

**NOTE:** When you run a batch task with an object WHERE clause, only records specified by the object WHERE clause are read into memory. However, depending on the number of records and customization, a single task can still consume a large amount of memory. To limit the total amount of memory used by the Data Quality Object Manager for concurrent tasks, you can reduce the value of the MaxTasks server component parameter setting so that fewer concurrent tasks run. For more information about setting the MaxTasks parameter, see *Applications Administration Guide*.



- After your initial data matching or key generation, include only new and updated records in your key generation and data matching processes because reprocessing all records is too time consuming.

You use the DeDup Key Modification Date and DeDup Last Match Date business component fields in your search specifications to exclude records. For example, the following table shows the Object WHERE Clause to run key generation or data matching.

To Query for...	Key Generation Example	Data matching Example
Updated records	([[DeDup Key Modification Date]<[Updated]])	([[DeDup Last Match Date]<[Updated]])
New records	([[DeDup Key Modification Date] IS NULL])	([[DeDup Last Match Date] IS NULL])
Updated and new records	([[DeDup Key Modification Date]<[Updated]]) OR ([[DeDup Key Modification Date] IS NULL])	(([[DeDup Last Match Date]<[Updated]]) OR ([[DeDup Last Match Date] IS NULL]))

- Set the object sort clause based on the fields that are used to generate match keys:
  - For person (contact or prospect), use Last Name, First Name, Middle Name.
  - For company (account), use Name or Name, Location.
- Set the DQSetting parameter to Delete to improve the performance of batch data matching and key generation processing.

By default, when you run data matching using SSA-NAME3, existing duplicate records are not removed from the S\_DEDUP\_RESULT table. Likewise when you run key generation batch jobs, existing keys are not removed.

To remove all keys in the key tables or all duplicate records in the S\_DEDUP\_RESULT table, run the appropriate batch job with DQSetting = Delete.

**NOTE:** The Delete setting is an *optional* Data Quality Setting parameter, whereas BCName, BObjName, and OpType are *required*.

**CAUTION:** Do not attempt to use the Delete option if you are not an expert user of SQL as you run the risk of corrupting your database.

For more information about running batch key generation jobs, see [“Generating or Refreshing Keys Using Batch Jobs” on page 51](#).

For more information about running batch data matching jobs, see [“Customizing Data Quality Server Component Jobs for Batch Mode” on page 56](#).

## Data Quality Settings

The following are recommended values for the settings in the Administration - Data Quality > Data Quality Settings view:

- **Key Type.** Set to Limited.
- **Match Threshold.** Set to a number greater than or equal to 75. The higher the threshold, the faster the data matching process runs.
- **Search Type.** Set to Narrow.

For more information about the Data Quality settings, see [“Specifying Data Quality Settings” on page 37](#).

# A

## Invoking Siebel Data Quality from External Callers

This appendix provides information about calling data matching and data cleansing methods from scripts or workflows. It includes the following topics:

- [“Invocation of SDQ from External Callers” on page 99](#)
- [“Scenario for Data Matching Using the Value Match Method” on page 99](#)
- [“Value Match Method” on page 100](#)
- [“Scenario for Data Cleansing Using Data Cleansing Business Service Methods” on page 103](#)
- [“Get Siebel Fields Method” on page 104](#)
- [“Parse Method” on page 104](#)
- [“Parse All Method” on page 105](#)

### Invocation of SDQ from External Callers

You can invoke SDQ from external callers to perform data matching. You can use the Value Match method of the DeDuplication business service to:

- Match data in field/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 invoke SDQ from external callers to perform data cleansing. There are three preconfigured Data Cleansing business service methods—Get Siebel Fields, Parse, and Parse All. Using an external caller, such as scripting or a workflow process, you first invoke the Get Siebel Fields method, and then invoke the Parse method to cleanse contacts and accounts.

The Parse All method is used automatically during data cleansing batch jobs, but calling Parse All from an external caller is not supported for this release.

### Scenario for Data Matching Using the Value Match Method

This topic gives one example of how you can invoke 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 needs to 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 invokes the Value Match method.

In this case, the implementation invokes 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 based on 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 invoke and use the Value Match method, see [“Value Match Method” on page 100](#).

## 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.

### Arguments

The Value Match method arguments consist of input and output property sets. [Table 27](#) provides the input property sets, and [Table 28 on page 101](#) provides the output property sets.

**CAUTION:** The Value Match property sets are specialized. Do not configure these components.

Table 27. Value Match Input Property Sets

Input Property Set	Property	Description	Comments
Adapter Settings <sup>1</sup>	Code Page	Code page.	Optional. Applicable only to SSA.
	Population	Population values.	The value Override can be specified to override the corresponding setting information obtained by the service from the administration screens, user properties, and so on.
	Search Level	The search level.	
	Threshold	The threshold score for a duplicate record. A match is considered only if the score exceeds this value.	

Table 27. Value Match Input Property Sets

Input Property Set	Property	Description	Comments
Generic Settings	BC Name	The name of the matched business component.	Required.
	Update Modification Date	If set to N, the match modification date is not updated.	Optional. Default = Y.
	Use Result Table	If set to N, matches are not added to the result table. Instead, matches are determined by the business service.	Optional. Default = Y.
Match Values <sup>1</sup>	<i>business component field names, value pairs</i>	The matched business component's field name and the corresponding field value. For example:  (Last Name, 'Smith') (First Name, 'John') and so on  <b>NOTE:</b> 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 user properties of the matched business component is used to map the business component field names to vendor field names.

1. 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 Table 28.

**CAUTION:** The Value Match property sets are specialized. Do not configure these components.

Table 28. Value Match Output Property Sets

Output Property Set <sup>1</sup>	Property	Description	Comments
Match Info	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.	

Table 28. Value Match Output Property Sets

Output Property Set <sup>1</sup>	Property	Description	Comments
Generic Settings	End Time	The run end time.	Applicable only to SSA.
	Num Candidates	The total number of potential matches if scores are not used.	
	Num Results	The number of actual matches.	
	Row Value	The row ID of the match or matches found.	
	Start Time	The run start time.	

1. Match Info is a child property set of the output property set.

**Invoked From**

Any means by which you can invoke business service methods, such as with Siebel eScript or from a workflow process.

**Example**

The following is an example of using Siebel eScript to invoke 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();
}
```

#### See Also

For more information about business services and methods, see *Siebel Developer's Reference*.

## Scenario for Data Cleansing Using Data Cleansing Business Service Methods

This topic gives one example of how you can invoke 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 invokes the Get Siebel Fields Method, which returns a list of cleansed fields for the applicable business component. Then the script or workflow invokes the Parse method, which returns the data for the cleansed fields.

## 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.

### Arguments

Get Siebel Fields arguments are listed in [Table 29](#).

Table 29. Get Siebel Fields Arguments

Argument Name	Display Name	Input/Output	Data Type	Req'd?	Description
BusComp Name	Bus Comp Name	Input	String	No	The name of the business component.
Field Names	Field Names	Output	Hierarchy	Yes	The name of the hierarchy.

### 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 method in the process of using a batch job to cleanse data.

### Invoked From

Any means by which you can invoke business service methods, such as with Siebel Workflow or Siebel eScript.

### See Also

For more information about business services and methods, see *Siebel Developer's Reference*.

## Parse Method

Parse is one of the methods of the Data Cleansing business service. This method returns the cleansed field data.



**Arguments**

Parse arguments are listed in [Table 30](#).

Table 30. Parse Arguments

Argument Name	Display Name	Input/Output	Data Type	Req'd?	Description
BusComp Name	Bus Comp Name	Input	String	No	The name of the business component.
Input Field Values	Input Field Values	Input	Hierarchy	Yes	A list of field values.
Output Field Values	Output Field Values	Output	Hierarchy	Yes	A list of field values.

**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.

**Invoked From**

Any means by which you can invoke business service methods, such as with Siebel Workflow or Siebel eScript.

**See Also**

For more information about Siebel Workflow, see *Siebel Business Process Designer Administration Guide*. For more information about business services and methods, see *Siebel Developer's Reference*.

## Parse All Method

Parse All is one of the methods of the Data Cleansing business service. This method cleanses the fields of a business component.

**NOTE:** The Parse All method is used automatically during data cleansing batch jobs, but calling Parse All from an external caller is not supported for this release.



# B

## Preconfigured Matching Server Properties and Field Mapping Values

This appendix lists preconfigured data matching property values for the Siebel Data Quality (SDQ) Matching Server using Search Software America (SSA) software. It includes the following topics:

- ["Preconfigured Matching Server Properties" on page 107](#)
- ["Preconfigured SSA Field Mappings" on page 108](#)

### Preconfigured Matching Server Properties

Matching Server definitions are configured as DeDuplication Business Service user properties. The SDQ Matching Server is preconfigured for Search Software America (SSA) for data matching.

#### *To view the preconfigured data matching properties for SSA*

- 1 Start Siebel Tools.
- 2 In the Object Explorer, expand the Business Service object, and then select DeDuplication.
- 3 In the Object Explorer, select the Business Service User Prop object.

**TIP:** If the Business Service 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).

Table 31 lists the preconfigured data matching properties for SSA.

Table 31. Preconfigured Business Service User Properties

Property Name	Value
SSA DLL	n3sgsb
SSA Default Universal Setting Key Type	Standard
SSA Default Universal Setting Match Level	Typical
SSA Default Universal Setting Search Type	Typical
SSA Default Universal Setting Threshold	75
SSA Population-Codepage CHS	"China", "Chinese_Simp"
SSA Population-Codepage CHT	"China", "Chinese_Trad"
SSA Population-Codepage CSY	"Czech", "Latin_2_1250"

Table 31. Preconfigured Business Service User Properties

Property Name	Value
SSA Population-Codepage DAN	"Denmark", "Latin_1_Mixed"
SSA Population-Codepage DEU	"Germany", "Latin_1_Mixed"
SSA Population-Codepage ELL	"Greece", "Greek"
SSA Population-Codepage ENU	"Default", "Latin_1_Mixed"
SSA Population-Codepage ESN	"Spain", "Latin_1_Mixed"
SSA Population-Codepage FIN	"Finland", "Latin_1_Mixed"
SSA Population-Codepage FRA	"France", "Latin_1_Mixed"
SSA Population-Codepage HEB	"Israel", "Hebrew"
SSA Population-Codepage ITA	"Italy", "Latin_1_Mixed"
SSA Population-Codepage JPN	"Japan", "Japanese"
SSA Population-Codepage KOR	"South_Korea", "Korean"
SSA Population-Codepage NLD	"Netherlands", "Latin_1_Mixed"
SSA Population-Codepage PLK	"Poland", "Latin_2_1250"
SSA Population-Codepage PSL	"Default", "Latin_1_Mixed"
SSA Population-Codepage PTB	"Brazil", "Latin_1_Mixed"
SSA Population-Codepage PTG	"Portugal", "Latin_1_Mixed"
SSA Population-Codepage SVE	"Sweden", "Latin_1_Mixed"
SSA Population-Codepage THA	"Thailand", "Thai"

## Preconfigured SSA Field Mappings

The field mappings from vendor fields to Siebel fields are stored in configurable user properties for the relevant business components. The following topics provide tables of the field mappings for SSA applications as well as other user properties for the business components.

For information about mapping fields for data matching, including syntax, see ["Mapping of Connector Fields to Business Component Fields" on page 67](#).

### Viewing Preconfigured SSA Field Mappings

Use the following procedure to view the preconfigured field mappings for SSA applications.

*To view preconfigured SSA field mappings*

- 1 Start Siebel Tools.
- 2 In the Object Explorer, expand the Business Component object.
- 3 In the Business Objects list, select the business component.  
For example, select Account.
- 4 In the Object Explorer, click Business Component User Prop.

## Preconfigured Account Business Component Field Mappings

Table 32 shows the data matching field mappings for the Account business component.

Some of the input field values are indicated by a lettering nomenclature where different letters indicate standard input types for personal name, company name, address fields, and ID data. For example, Z indicates postal or ZIP code while I indicates a general unique identifier such as the D-U-N-S number for accounts or social security number for contacts. For more information about field mappings for business components using the embedded SSA-NAME3 software, see the third-party documentation for SSA-NAME3 on the *Siebel Business Applications Third-Party Bookshelf*.

Table 32. Preconfigured SSA Data Matching Mappings for the Account Business Component

User Property Name	Value
DeDuplication Connector - SSA	"SSA"
Dedup Type	Account
SSA DeDuplication Field 1	"C", "Name"
SSA DeDuplication Field 2	"Country", "Country"
SSA DeDuplication Field 3	"City", "City"
SSA DeDuplication Field 4	"State", "State"
SSA DeDuplication Field 5	"Z", "Postal Code"
SSA DeDuplication Field 6	"I", "DUNS Number"
SSA DeDuplication Field 7	"A", "Street Address"

## Preconfigured Contact Business Component User Properties

Table 33 shows the data matching field mappings for the Contact business component.

Table 33. Preconfigured SSA Data Matching Mappings for the Contact Business Component

User Property Name	Value
DeDuplication Connector - SSA	"SSA"
Dedup Type	Contact
SSA DeDuplication Field 1	"Z", "Postal Code"
SSA DeDuplication Field 2	"T", "Work Phone #"
SSA DeDuplication Field 3	"T", "Cellular Phone #"
SSA DeDuplication Field 4	"I", "Social Security Number"
SSA DeDuplication Field 5	"E", "Email Address"
SSA DeDuplication Field 6	"A", "Street Address"
SSA DeDuplication Field 7	"T", "Home Phone #"
SSA DeDuplication Field 8	"Last", "Last Name"
SSA DeDuplication Field 9	"First", "First Name"
SSA DeDuplication Field 10	"Middle", "Middle Name"
SSA DeDuplication Field 11	"City", "City"
SSA DeDuplication Field 12	"State", "State"
SSA DeDuplication Field 13	"Country", "Country"
SSA DeDuplication Field 14	"T", "Home Phone #"
SSA DeDuplication Field 15	"D", "Birth Date"

## Preconfigured List Mgmt Prospective Contact Business Component Field Mappings

Table 34 shows the data matching field mappings for the List Mgmt Prospective Contact business component.

Table 34. Preconfigured SSA Data Matching Mappings for the List Mgmt Prospective Contact Business Component

User Property Name	Value
DeDuplication Connector - SSA	"SSA"
Dedup Type	List Mgmt Prospective Contact
SSA DeDuplication Field 1	"Last", "Last Name"
SSA DeDuplication Field 2	"First", "First Name"
SSA DeDuplication Field 3	"Middle", "Middle Name"
SSA DeDuplication Field 4	"C", "Account"
SSA DeDuplication Field 5	"A", "Street Address"
SSA DeDuplication Field 6	"City", "City"
SSA DeDuplication Field 7	"State", "State"
SSA DeDuplication Field 8	"Z", "Postal Code"
SSA DeDuplication Field 9	"Country", "Country"
SSA DeDuplication Field 10	"T", "Work Phone #"
SSA DeDuplication Field 11	"T", "Cellular Phone #"
SSA DeDuplication Field 12	"I", "Social Security Number"
SSA DeDuplication Field 13	"E", "Email Address"
SSA DeDuplication Field 14	"T", "Home Phone #"





# C

## Universal Connector API

This appendix describes the application programming interface (API) functions that third-party software vendors must implement in the dynamic link libraries (DLL) that they provide for use with the SDQ Universal Connector. It includes the following topics:

- [“Vendor Dynamic Link Libraries” on page 113](#)
- [“Connector Initialization and Termination Functions” on page 114](#)
- [“Session Initialization and Termination Functions” on page 115](#)
- [“Parameter Setting Functions” on page 116](#)
- [“Error Message Functions” on page 119](#)
- [“Real-Time Data Matching Functions” on page 119](#)
- [“Batch Mode Data Matching Functions” on page 122](#)
- [“Real-Time Data Cleansing Functions” on page 128](#)
- [“Batch Mode Data Cleansing Functions” on page 129](#)
- [“Data Matching and Data Cleansing Algorithms” on page 130](#)

## Vendor Dynamic Link Libraries

Vendors must follow these rules for their DLLs:

- The DLLs must be thread-safe. A DLL can support multiple sessions by using different unique session IDs.
- The DLLs must support UTF-16 (UCS2) as the default Unicode encoding.
- If there is a single DLL for all supported languages, the DLLs must be named as follows:
  - *BASE.dll* (on Windows)
  - *libBASE.so* (on AIX and Solaris)
  - *libBASE.sl* (on HP-UX)

where *BASE* is a name chosen by the vendor. If a vendor has many solutions for different types of data, they can use different base names for different DLLs.

- If there are separate DLLs for different languages, the DLL name must include the appropriate language code. For example, for Japanese (JPN), the DLLs must be named as follows:
  - *BASEjpn.dll* (on Windows)
  - *libBASEjpn.so* (on AIX and Solaris)
  - *libBASEjpn.sl* (on HP-UX)

The Siebel application loads the DLLs from the locations described in [Table 8 on page 32](#).

The mapping of Siebel 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 DLL must be stored outside of Siebel.

## Terminology

The following terms are used in this appendix:

- **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.
- **Master record.** The record for which data matching was performed.

# Connector Initialization and Termination Functions

This topic describes functions that are called when the vendor DLL is loaded or when the Siebel Server shuts down.

## sdq\_init\_connector Function

This function is called when the vendor DLL is first loaded to facilitate any initialization tasks.

### Syntax

```
int sdq_init_connector (const SSchar * path)
```

### Parameters

Parameter	Description
path	The absolute path of the Siebel Server installation. The vendor 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.

**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.

## **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.

**Syntax**

```
int sdq_init_session (int * session_id)
```

## Parameters

Parameter	Description
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 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.

### Syntax

```
int sdq_close_session (int * session_id)
```

## Parameters

Parameter	Description
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 DLL'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

This function is called to set global parameters. This function call is made after the call to `sdq_init_connector`.

Before making the call to the vendor DLL, the Siebel code reads the parameters and values from the configuration file that is read upon server or remote client startup. Putting parameters here enables customers to set and modify connector and server-instance specific parameters.

An XML character string is used to specify the parameters. This provides an extensible way of providing parameters with each function call.

**NOTE:** For the 7.8.2 release, this parameter is set to NULL as all required parameters are set by the `sdq_set_parameter` function call.

### Syntax

```
int sdq_set_global_parameter (const SSchar* parameterList)
```

### Parameters

Parameter	Description
parameterList	<p>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</p> <pre>&lt;Data&gt;   &lt;Parameter&gt;     &lt;Name&gt;Global ConnectorParam1&lt;/Name&gt;     &lt;Value&gt;Global ConnectorValue1&lt;/Value&gt;   &lt;/Parameter&gt;   &lt;Parameter&gt;     &lt;Name&gt;Global ConnectorParam2&lt;/Name&gt;     &lt;Value&gt;Global ConnectorValue2&lt;/Value&gt;   &lt;/Parameter&gt; &lt;/Data&gt;</pre>

### 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.

Before making the call to the vendor DLL, the Siebel code reads the parameters and values from the configuration file that is read upon server or remote client startup. Putting parameters here enables customers to set and modify connector and server-instance specific parameters.

### Syntax

```
int sdq_set_parameter (int session_id, const SSchar* parameterList)
```

### Parameters

Parameter	Description
session_id	The session ID obtained while initializing the session.
parameterList	<p>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:</p> <pre>&lt;Data&gt;     &lt;Parameter&gt;         &lt;Name&gt;SessionParam1&lt;/Name&gt;         &lt;Value&gt;SessionValue1&lt;/Value&gt;     &lt;/Parameter&gt;     &lt;Parameter&gt;         &lt;Name&gt;SessionParam2&lt;/Name&gt;         &lt;Value&gt;SessionValue2&lt;/Value&gt;     &lt;/Parameter&gt; &lt;/Data&gt;</pre>

### 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 Functions

This topic describes the function associated with error messages.

### 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.

#### Syntax

```
void sdq_get_error_message (int error_code, SSchar * error_summary, SSchar * error_details)
```

#### Parameters

Parameter	Description
error_code	The error code returned from the previous function call.
error_summary	A pointer to the error message summary, which is up to 256 characters long.
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 functions that are called for real-time data matching.

### sdq\_dedup\_realtime Function

This function is called to perform real-time data matching.

This function send the data for each record as driver records and their candidate records. The function is called only once; multiple calls to the vendor DLL 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.

**Syntax**

int sdq\_dedup\_realtime (int session\_id, SSchar\* parameterList, char\* inputRecordSet, char\* outputRecordSet)

**Parameters**

Parameter	Description
session_id	The session ID obtained by initializing the session.
parameterList	<p>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:</p> <pre> &lt;Data&gt;   &lt;Parameter&gt;     &lt;Name&gt;Real TimeDedupParam1&lt;/Name&gt;     &lt;Value&gt;Real TimeDedupValue1&lt;/Value&gt;   &lt;/Parameter&gt;   &lt;Parameter&gt;     &lt;Name&gt;Real TimeDedupParam2&lt;/Name&gt;     &lt;Value&gt;Real TimeDedupValue2&lt;/Value&gt;   &lt;/Parameter&gt; &lt;/Data&gt; </pre> <p><b>NOTE:</b> For the 7.8.2 release, this parameter is set to NULL as all required parameters are already set at the session level.</p>



Parameter	Description
InputRecordSet	<p>An XML character string containing the driver record and candidate records. An example of the XML is as follows:</p> <pre data-bbox="587 449 1411 1318"> &lt;Data&gt;   &lt;DriverRecord&gt;     &lt;Account. Id&gt;1-X42&lt;/Account. Id&gt;     &lt;Account. Name&gt;Siebel &lt;/Account. Name&gt;     &lt;Account. Location&gt;Headquarters&lt;/Account. Location&gt;   &lt;/DriverRecord&gt;   &lt;CandidateRecord&gt;     &lt;Account. Id&gt;1-Y28&lt;/Account. Id&gt;     &lt;Account. Name&gt;Siebel &lt;/Account. Name&gt;     &lt;Account. Location&gt;Atlanta&lt;/Account. Location&gt;   &lt;/CandidateRecord&gt;   &lt;CandidateRecord&gt;     &lt;Account. Id&gt;1-3-P&lt;/Account. Id&gt;     &lt;Account. Name&gt;Siebel &lt;/Account. Name&gt;     &lt;Account. Location&gt;Rome&lt;/Account. Location&gt;   &lt;/CandidateRecord&gt; &lt;/Data&gt; </pre>

Parameter	Description
outputRecordSet	<p>An XML character string populated by the vendor in real time that contains the duplicate records with the scores. An example of the XML is as follows:</p> <pre> &lt;Data&gt;   &lt;DuplicateRecord&gt;     &lt;Account.Id&gt;1-Y28&lt;/Account.Id&gt;     &lt;DQ.MatchScore&gt;92&lt;/DQ.MatchScore&gt;   &lt;/DuplicateRecord&gt;   &lt;DuplicateRecord&gt;     &lt;Account.Id&gt;1-3-P&lt;/Account.Id&gt;     &lt;DQ.MatchScore&gt;88&lt;/DQ.MatchScore&gt;   &lt;/DuplicateRecord&gt; &lt;/Data&gt; </pre>

### 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

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.

#### Syntax

```
int sdq_set_dedup_candidates (int session_id, SSchar* parameterList, SSchar* xmlRecordSet)
```

## Parameters

Parameter	Description
session_id	The session ID obtained by initializing the session.
parameterList	<p>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:</p> <pre> &lt;Data&gt;   &lt;Parameter&gt;     &lt;Name&gt;BatchDedupParam1&lt;/Name&gt;     &lt;Value&gt;BatchDedupValue1&lt;/Value&gt;   &lt;/Parameter&gt;   &lt;Parameter&gt;     &lt;Name&gt;BatchDedupParam2&lt;/Name&gt;     &lt;Value&gt;BatchDedupValue2&lt;/Value&gt;   &lt;/Parameter&gt; &lt;/Data&gt; </pre> <p><b>NOTE:</b> For the 7.8.2 release, this parameter is set to NULL as all required parameters are already set at the session level.</p>

Parameter	Description
xmlRecordSet For full data matching batch jobs	<p>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:</p> <pre data-bbox="565 478 1422 1352"> &lt;Data&gt;   &lt;CandidateRecord&gt;     &lt;Account.Id&gt;2-24-E&lt;/Account.Id&gt;     &lt;Account.Name&gt;Siebel&lt;/Account.Name&gt;     &lt;Account.Location&gt;Somewhere&lt;/Account.Location&gt;   &lt;/CandidateRecord&gt;   &lt;CandidateRecord&gt;     &lt;Account.Id&gt;1-E-2E&lt;/Account.Id&gt;     &lt;Account.Name&gt;Siebel&lt;/Account.Name&gt;     &lt;Account.Location&gt;Somewhere else&lt;/Account.Location&gt;   &lt;/CandidateRecord&gt;   &lt;CandidateRecord&gt;     &lt;Account.Id&gt;2-34-F&lt;/Account.Id&gt;     &lt;Account.Name&gt;Siebel&lt;/Account.Name&gt;     &lt;Account.Location&gt;Someplace&lt;/Account.Location&gt;   &lt;/CandidateRecord&gt; &lt;/Data&gt; </pre>

Parameter	Description
xmlRecordSet For incremental data matching batch jobs	<p>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 need to return duplicate records:</p> <pre> &lt;Data&gt;   &lt;DriverRecord&gt;     &lt;Account.Id&gt;2-24-E&lt;/Account.Id&gt;     &lt;Account.Name&gt;Siebel &lt;/Account.Name&gt;     &lt;Account.Location&gt;Somewhere&lt;/Account.Location&gt;   &lt;/DriverRecord&gt;   &lt;CandidateRecord&gt;     &lt;Account.Id&gt;1-E-9E&lt;/Account.Id&gt;     &lt;Account.Name&gt;Siebel &lt;/Account.Name&gt;     &lt;Account.Location&gt;Somewhere else&lt;/Account.Location&gt;   &lt;/CandidateRecord&gt;   &lt;DriverRecord&gt;     &lt;Account.Id&gt;1-E-2E&lt;/Account.Id&gt;     &lt;Account.Name&gt;Siebel &lt;/Account.Name&gt;     &lt;Account.Location&gt;Somewhere else&lt;/Account.Location&gt;   &lt;/DriverRecord&gt;   &lt;CandidateRecord&gt;     &lt;Account.Id&gt;1-12-2H&lt;/Account.Id&gt;     &lt;Account.Name&gt;Siebel &lt;/Account.Name&gt;     &lt;Account.Location&gt;Somewhere else&lt;/Account.Location&gt;   &lt;/CandidateRecord&gt;   &lt;DriverRecord&gt;     &lt;Account.Id&gt;2-34-F&lt;/Account.Id&gt;     &lt;Account.Name&gt;Siebel &lt;/Account.Name&gt;     &lt;Account.Location&gt;Someplace&lt;/Account.Location&gt;   &lt;/DriverRecord&gt; &lt;/Data&gt;                     </pre>

**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.

**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.

**Syntax**

`int sdq_start_dedup (int session_id)`

**Parameters**

Parameter	Description
<code>session_id</code>	The session ID obtained by initializing the session.

## sdq\_get\_duplicates Function

This function is called to get the master record with the list of its duplicate records along with their match scores. This is done in batch mode. The number of records received per call to this function is set in the `BATCH_MATCH_MAX_NUM_OF_RECORDS` session parameter before the function is called.

**Syntax**

`int sdq_get_duplicates (int session_id, char* xmlResultSet)`

Parameters

Parameter	Description
session_id	The session ID obtained by initializing the session.
xmlRecordSet	<p>An XML character string that the vendor DLL populates with a master record and a list of its duplicate records along with their match scores.</p> <p>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 master record as well. The XML is in the following format:</p> <pre> &lt;Data&gt;   &lt;ParentRecord&gt;     &lt;DQ.MasterRecordsRowID&gt;2-24-E&lt;/DQ.MasterRecordsRowID&gt;     &lt;DuplicateRecord&gt;       &lt;Account.ID&gt;2-24-E&lt;/Account.ID&gt;       &lt;DQ.MatchScore&gt;92&lt;/DQ.MatchScore&gt;     &lt;/DuplicateRecord&gt;     &lt;DuplicateRecord&gt;       &lt;Account.ID&gt;2-23-F&lt;/Account.ID&gt;       &lt;DQ.MatchScore&gt;88&lt;/DQ.MatchScore&gt;     &lt;/DuplicateRecord&gt;   &lt;/ParentRecord&gt; &lt;/Data&gt; </pre>

Return Value

A return value of 0 indicates successful execution, while a return value of 1 indicates that there are no duplicate records left. 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.

**NOTE:** Siebel 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.

# Real-Time Data Cleansing Functions

This topic describes the functions that are called for real-time data matching.

## sdq\_datacleanse Function

This function is called to perform real-time data cleansing. The function is called for only one record at a time.

### Syntax

```
int sdq_datacleanse (int session_id, SSchar* parameterList, SSchar* inputRecordSet, SSchar* outputRecordSet)
```

### Parameters

Parameter	Description
session_id	A session ID obtained by initializing the session.
parameterList	<p>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:</p> <pre>&lt;Data&gt;   &lt;Parameter&gt;     &lt;Name&gt;Real TimeDataCleansseParam1&lt;/Name&gt;     &lt;Value&gt;Real TimeDataCleansseValue1&lt;/Value&gt;   &lt;/Parameter&gt;   &lt;Parameter&gt;     &lt;Name&gt;Real TimeDataCleansseParam2&lt;/Name&gt;     &lt;Value&gt;Real TimeDataCleansseValue2&lt;/Value&gt;   &lt;/Parameter&gt; &lt;/Data&gt;</pre> <p><b>NOTE:</b> For the 7.8.2 release, this parameter is set to NULL as all required parameters are already set at the session level.</p>



Parameter	Description
inputRecordSet	An XML character string containing the driver record. An example of the XML is as follows: <pre>                     &lt;Data&gt;                         &lt;DriverRecord&gt;                             &lt;Contact.Fi rstName&gt;mi chael &lt;/Contact.Fi rstName&gt;                             &lt;Contact.LastName&gt;mouse&lt;/Contact.LastName&gt;                         &lt;/DriverRecord&gt;                     &lt;/Data&gt;                     </pre>
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: <pre>                     &lt;Data&gt;                         &lt;CleansedDriverRecord&gt;                             &lt;Contact.Fi rstName&gt;Mi chael &lt;/Contact.Fi rstName&gt;                             &lt;Contact.LastName&gt;Mouse&lt;/Contact.LastName&gt;                         &lt;/CleansedDriverRecord&gt;                     &lt;/Data&gt;                     </pre>

### 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 Functions

This topic describes the functions that are called for batch mode data cleansing.

### sdq\_data\_cleanse Function

The same function, is called by Siebel 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 code for both real-time and batch data cleansing and data matching,

## Batch Data Matching Algorithm

The algorithm is as follows:

- 1 Load the vendor DLL.
- 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. By default there is only one match key generated, but if needed, multiple keys can be supported as well. For more information about match key generation, see [“Generating or Refreshing Keys Using Batch Jobs” on page 51](#).

- 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 master 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 DLL.
- 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 Query the Siebel database to get the Candidate records for the driver record.
- 7 Call `sdq_dedup_realtime`.
- 8 Call `sdq_close_session` while logging out of current session.
- 9 Call `sdq_close_connector`.

## Batch Data Cleansing Algorithm

The algorithm is as follows:

- 1 Load the vendor DLL.
- 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).
- 6 Query the Siebel database to get the set of records to be cleansed.
- 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 DLL.
- 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 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`.



# D

## Example of Universal Connector Properties and Field Mapping Values

This appendix lists an example of preconfigured data cleansing and data matching property values for the Siebel Data Quality (SDQ) Universal Connector using third-party software. The definitions in this appendix are as preconfigured for Firstlogic software.

This appendix includes the following topics:

- [“Preconfigured Connector Properties” on page 133](#)
- [“Preconfigured Field Mappings” on page 135](#)

### Preconfigured Connector Properties

Connector definitions are preconfigured as DeDuplication business service and Data Cleansing business service user properties.

Two sets of properties are given in separate subtopics of this topic:

- Properties applicable from the 7.8.2.2 release and later.
- Properties applicable up to the 7.8.2 release.

**CAUTION:** Do not reconfigure the property settings.

### Viewing Preconfigured Connector Properties

Use the following procedure to view the preconfigured connector properties.

#### *To view the preconfigured connector properties*

- 1 Start Siebel Tools.
- 2 In the Object Explorer, expand the Business Service object, and then select DeDuplication or Data Cleansing
- 3 In the Object Explorer, select the Business Service User Prop object.

**TIP:** If the Business Service 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).

### Preconfigured Business Service Properties (7.8.2.2)

This topic lists the preconfigured Business Service properties valid for the 7.8.2.2 release and later.

## Preconfigured DeDuplication Business Service User Properties

Table 35 lists the DeDuplication business service user properties preconfigured for Firstlogic.

Table 35. Preconfigured DeDuplication Business Service User Properties

Property Name	Value
Firstlogic Connector 1	Dedup
Dedup Attribute - Library Base Name	FirstlogicConnector

## Preconfigured Data Cleansing Business Service User Properties

Table 36 lists the Data Cleansing business service user properties preconfigured for Firstlogic.

Table 36. Preconfigured Data Cleansing Business Service User Properties

Property Name	Value
Firstlogic Connector 1	DataCleanse
DataCleanse Attribute - Library Base Name	FirstlogicConnector

**NOTE:** From release 7.8.2.2, the Firstlogic Connector 1 property maps the Data Cleansing business service to the Firstlogic library, FirstlogicConnector.dll.

## Preconfigured Business Service Properties (7.8.2)

This topic lists the preconfigured Business Service properties valid up to the 7.8.2 release.

## Preconfigured DeDuplication Business Service User Properties

Table 37 lists some of the DeDuplication business service user properties preconfigured for Firstlogic.

**NOTE:** The property values with three strings show type of parameter, parameter name, and parameter value, in that order.

Table 37. Preconfigured DeDuplication Business Service User Properties

Property Name	Value
Vendor1 Connector 1	match
match Attribute - Library Base Name	sdqmatch
match Parameter 1	"global", "MATCH_FIELD_LASTNAME", "50"
match Parameter 2	"global", "MATCH_FIELD_FIRSTNAME", "50"
match Parameter 3	"global", "MATCH_FIELD_MIDDLENAME", "50"

Table 37. Preconfigured DeDuplication Business Service User Properties

Property Name	Value
match Parameter 4	"global", "MATCH_FIELD_FIRM", "100"
match Parameter 5	"global", "MATCH_FIELD_FIRMLOC", "50"
match Parameter 6	"global", "MATCH_FIELD_UNPADDRLINE", "200"
match Parameter 7	"global", "MATCH_FIELD_CITY", "50"
match Parameter 8	"global", "MATCH_FIELD_STATE", "10"
match Parameter 9	"global", "MATCH_FIELD_ZIP4", "30"
match Parameter 10	"global", "MATCH_FIELD_COUNTRY", "30"

## Preconfigured Data Cleansing Business Service User Properties

Table 38 lists some of the Data Cleansing business service user properties preconfigured for Firstlogic.

Table 38. Preconfigured DeDuplication Business Service User Properties

Property Name	Value
Vendor1 Connector 1	namefirm
Vendor1 Connector 2	address
address Attribute - Library Base Name	sdqaddress
namefirm Attribute - Library Base Name	sdqname

**NOTE:** Up to release 7.8.2, The Connector 1 and Connector 2 properties map the Data Cleansing business service to two distinct Firstlogic libraries, sdqaddress.dll and sdqname.dll

## Preconfigured Field Mappings

The field mappings from vendor fields to Siebel fields are stored in configurable user properties for the relevant business components. The following topics provide tables of the field mappings for Firstlogic applications as well as other user properties for the business components.

Two sets of field mappings are given in separate subtopics of this topic:

- Mappings applicable from the 7.8.2.2 release.
- Mappings applicable up to the 7.8.2 release.

For information about mapping fields for data cleansing, including syntax, see ["Mapping Data Cleansing Connector Fields to Siebel Business Component Fields"](#) on page 71

For information about mapping fields for data matching, including syntax, see ["Mapping of Connector Fields to Business Component Fields"](#) on page 67.

## Viewing Preconfigured Field Mappings

Use the following procedure to view the preconfigured field mappings for Firstlogic applications.

### To view preconfigured field mappings

- 1 Start Siebel Tools.
- 2 In the Object Explorer, expand the Business Component object.
- 3 In the Business Objects list, select the business component.  
For example, select Account.
- 4 In the Object Explorer, click Business Component User Prop.

## Preconfigured Business Component User Properties (7.8.2.2)

This topic lists the preconfigured Business Service properties valid for the 7.8.2.2 release and later.

### Preconfigured Account Business Component User Properties

Table 39 shows data matching user properties for the Account business component.

Table 39. Preconfigured Properties for the Account Business Component

User Property Name	Value
DeDuplication Connector - Firstlogic	"Dedup"
Dedup Type	Account
Dedup Token Expression	"IfNull (Left ([Primary Account Postal Code], 5), '____') + IfNull (Left ([Name], 1), '_') + IfNull (Mid ([Street Address], FindNoneOf ([Street Address], '1234567890 '), 1), '_")"
Dedup Query Expression	"IfNull (Left ([Primary Account Postal Code], 5), '????') + IfNull (Left ([Name], 1), '?') + IfNull (Mid ([Street Address], FindNoneOf ([Street Address], '1234567890 '), 1), '?")"
Dedup DeDuplication Field 1	"Account.Name", "Name"
Dedup DeDuplication Field 2	"Account.Location", "Location"
Dedup DeDuplication Field 3	"Account.Street Address", "Street Address"
Dedup DeDuplication Field 4	"Account.City", "City"
Dedup DeDuplication Field 5	"Account.State", "State"
Dedup DeDuplication Field 6	"Account.Postal Code", "Postal Code"



Table 39. Preconfigured Properties for the Account Business Component

User Property Name	Value
Dedup DeDuplication Field 7	"Account.Country", "Country"
Dedup DeDuplication Field 8	"Account.Id", "Id"
Dedup DeDuplication Field 9	"Account.MatchCandidateToken", "Dedup Token"

Table 40 shows data cleansing user properties for the Account business component

Table 40. Preconfigured Firstlogic Data Cleansing Properties for the Account Business Component

User Property Name	Value
DataCleansing Connector - Firstlogic	"DataCleanse"
DataCleanse Type	Account
DataCleanse DataCleansing Field 1	"Name", "Account.Name"
DataCleanse DataCleansing Field 2	"Location", "Account.Location"

## Preconfigured Contact Business Component User Properties

Table 41 shows data matching user properties for the Contact business component.

Table 41. Preconfigured Firstlogic Data Matching Properties for the Contact Business Component

User Property Name	Value
DeDuplication Connector - Firstlogic	"Dedup"
Dedup Type	Contact
Dedup Token Expression	"IfNull (Left ([Postal Code], 5), '_____') + IfNull (Left ([Account], 1), '_') + IfNull (Left ([Last Name], 1), '_')"
Dedup Query Expression	"IfNull (Left ([Postal Code], 5), '?????') + IfNull (Left ([Account], 1), '?') + IfNull (Left ([Last Name], 1), '?")"
Dedup DeDuplication Field 1	"Contact.First Name", "First Name"
Dedup DeDuplication Field 2	"Contact.Middle Name", "Middle Name"
Dedup DeDuplication Field 3	"Contact.Last Name", "Last Name"
Dedup DeDuplication Field 4	"Contact.Account", "Account"
Dedup DeDuplication Field 5	"Contact.Account Location", "Account Location"
Dedup DeDuplication Field 6	"Contact.Street Address", "Street Address"
Dedup DeDuplication Field 7	"Contact.City", "City"

Table 41. Preconfigured Firstlogic Data Matching Properties for the Contact Business Component

User Property Name	Value
Dedup DeDuplication Field 8	"Contact.State", "State"
Dedup DeDuplication Field 9	"Contact.Postal Code", "Postal Code"
Dedup DeDuplication Field 10	"Contact.Country", "Country"
Dedup DeDuplication Field 11	"Contact.Id", "Id"
Dedup DeDuplication Field 12	"Contact.MatchCandidateToken", "Dedup Token"

Table 42 shows the data cleansing user properties for the Contact business component.

Table 42. Preconfigured Firstlogic Data Cleansing Properties for the Contact Business Component

User Property Name	Value
DataCleansing Connector - Firstlogic	"DataCleanse"
DataCleanse Type	Contact
DataCleanse DataCleansing Field 1	"Last Name", "Contact.Last Name"
DataCleanse DataCleansing Field 2	"First Name", "Contact.First Name"
DataCleanse DataCleansing Field 3	"Middle Name", "Contact.Middle Name"
DataCleanse DataCleansing Field 4	"Job Title", "Contact.Job Title"

## Preconfigured List Mgmt Prospective Contact Business Component User Properties

Table 43 shows the data matching user properties for the List Mgmt Prospective Contact business component.

Table 43. Preconfigured Firstlogic Data Matching Properties for the List Mgmt Prospective Contact Business Component

User Property Name	Value
DeDuplication Connector - Firstlogic	"Dedup"
Dedup Type	List Mgmt Prospective Contact
Dedup Token Expression	"IfNull (Left ([Postal Code], 5), '_____') + IfNull (Left ([Account], 1), '_') + IfNull (Left ([Last Name], 1), '_)'"
Dedup Query Expression	"IfNull (Left ([Postal Code], 5), '?????') + IfNull (Left ([Account], 1), '?') + IfNull (Left ([Last Name], 1), '?)'"

Table 43. Preconfigured Firstlogic Data Matching Properties for the List Mgmt Prospective Contact Business Component

User Property Name	Value
Dedup DeDuplication Field 1	"List Mgmt Prospective Contact.First Name", "First Name"
Dedup DeDuplication Field 2	"List Mgmt Prospective Contact.Middle Name", "Middle Name"
Dedup DeDuplication Field 3	"List Mgmt Prospective Contact.Last Name", "Last Name"
Dedup DeDuplication Field 4	"List Mgmt Prospective Contact.Account", "Account"
Dedup DeDuplication Field 5	"List Mgmt Prospective Contact.Primary Account Location", " Primary Account Location"
Dedup DeDuplication Field 6	"List Mgmt Prospective Contact.Street Address", "Street Address"
Dedup DeDuplication Field 7	"List Mgmt Prospective Contact.City", "City"
Dedup DeDuplication Field 8	"List Mgmt Prospective Contact.State", "State"
Dedup DeDuplication Field 9	"List Mgmt Prospective Contact.Postal Code", "Postal Code"
Dedup DeDuplication Field 10	"List Mgmt Prospective Contact.Country", "Country"
Dedup DeDuplication Field 11	"List Mgmt Prospective Contact.Id", "Id"
Dedup DeDuplication Field 12	"List Mgmt Prospective Contact.MatchCandidateToken", "Dedup Token"

Table 44 shows the data cleansing user properties for the List Mgmt Prospective Contact business component.

Table 44. Preconfigured Firstlogic Data Cleansing Properties for the List Mgmt Prospective Contact Business Component

User Property Name	Value
DataCleansing Connector - Firstlogic	"DataCleanse"
DataCleanse Type	List Mgmt Prospective Contact
DataCleanse DataCleansing Field 1	"Street Address", "List Mgmt Prospective Contact.Street Address"
DataCleanse DataCleansing Field 2	"Account", "List Mgmt Prospective Contact.Account"

Table 44. Preconfigured Firstlogic Data Cleansing Properties for the List Mgmt Prospective Contact Business Component

User Property Name	Value
DataCleanse DataCleansing Field 3	"Primary Account Location", "List Mgmt Prospective Contact.Primary Account Location"
DataCleanse DataCleansing Field 4	"Street Address 2", "List Mgmt Prospective Contact.Street Address 2"
DataCleanse DataCleansing Field 5	"City", "List Mgmt Prospective Contact.City"
DataCleanse DataCleansing Field 6	"State", "List Mgmt Prospective Contact.State"
DataCleanse DataCleansing Field 7	"Postal Code", "List Mgmt Prospective Contact.Postal Code"
DataCleanse DataCleansing Field 8	"Country", "List Mgmt Prospective Contact.Country"
DataCleanse DataCleansing Field 9	"Last Name", " List Mgmt Prospective Contact.Last Name"
DataCleanse DataCleansing Field 10	"First Name", " List Mgmt Prospective Contact.First Name"
DataCleanse DataCleansing Field 11	"Middle Name", " List Mgmt Prospective Contact.Middle Name"
DataCleanse DataCleansing Field 12	"Job Title", " List Mgmt Prospective Contact.Job Title"

## Preconfigured Business Address Business Component User Properties

Table 45 shows the data cleansing user properties for the Business Address business component.

Table 45. Preconfigured Data Cleansing Properties for the Business Address Business Component

User Property Name	Value
DataCleansing Connector - Firstlogic	"DataCleanse"
DataCleanse Type	Business Address
DataCleanse DataCleansing Field 1	"Street Address", "Business Address.Street Address"
DataCleanse DataCleansing Field 2	"City", "Business Address.City"
DataCleanse DataCleansing Field 3	"State", "Business Address.State"
DataCleanse DataCleansing Field 4	"Postal Code", "Business Address.Postal Code"
DataCleanse DataCleansing Field 5	"Country", "Business Address.Country"
DataCleanse DataCleansing Field 6	"Street Address 2", "Business Address.Street Address 2"

## Preconfigured Business Component User Properties (7.8.2)

This topic lists the preconfigured Business Component properties valid up to the 7.8.2. release.

### Preconfigured Account Business Component User Properties

Table 46 shows data matching user properties for the Account business component

**CAUTION:** MP\_KEYFLD\_KEYMISC is used to store the Siebel ROW\_ID during the data matching process; do not configure this user property.

Table 46. Preconfigured Firstlogic Data Matching Properties for the Account Business Component

User Property Name	Value
DeDuplication Connector Vendor1	"match"
match Type	Account
match DeDuplication Field 1	"MATCH_FIELD_FIRM", "Name"
match DeDuplication Field 2	"MATCH_FIELD_FIRMLOC", "Location"
match DeDuplication Field 3	"MATCH_FIELD_UNPADDRLINE", "Street Address"
match DeDuplication Field 4	"MATCH_FIELD_CITY", "City"
match DeDuplication Field 5	"MATCH_FIELD_STATE", "State"
match DeDuplication Field 6	"MATCH_FIELD_ZIP4", "Postal Code"
match DeDuplication Field 7	"MATCH_FIELD_COUNTRY", "Country"
match DeDuplication Field 8	"MP_KEYFLD_KEYMISC", ""

Table 47 shows data cleansing user properties for the Account business component.

Table 47. Preconfigured Firstlogic Data Cleansing Properties for the Account Business Component

User Property Name	Value
DataCleansing Connector - Vendor1	"namefirm"
namefirm DataCleansing Field 1	"Name", "drl_ifirmname1", "drl_firm_std1"
namefirm DataCleansing Field 2	"Location", "drl_ifirmloc1", "drl_firm_loc_std1"

## Preconfigured Contact Business Component User Properties

Table 48 shows data matching user properties for the Contact business component.

**CAUTION:** MP\_KEYFLD\_KEYMISC is used to store the Siebel ROW\_ID during the data matching process; do not configure this user property.

Table 48. Preconfigured Firstlogic Data Matching Properties for the Contact Business Component

User Property Name	Value
DeDuplication Connector - Vendor1	"match"
match Type	Contact
match DeDuplication Field 1	"MATCH_FIELD_LASTNAME", "Last Name"
match DeDuplication Field 2	"MATCH_FIELD_FIRSTNAME", "First Name"
match DeDuplication Field 3	"MATCH_FIELD_MIDDLENAME", "Middle Name"
match DeDuplication Field 4	"MATCH_FIELD_FIRM", "Account"
match DeDuplication Field 5	"MATCH_FIELD_FIRMLOC", "Account Location"
match DeDuplication Field 6	"MATCH_FIELD_UNPADDRLINE", "Street Address"
match DeDuplication Field 7	"MATCH_FIELD_CITY", "City"
match DeDuplication Field 8	"MATCH_FIELD_STATE", "State"
match DeDuplication Field 9	"MATCH_FIELD_ZIP4", "Postal Code"
match DeDuplication Field 10	"MATCH_FIELD_COUNTRY", "Country"
match DeDuplication Field 11	"MP_KEYFLD_KEYMISC", ""

Table 49 shows the data cleansing user properties for the Contact business component.

Table 49. Preconfigured Firstlogic Data Cleansing Properties for the Contact Business Component

User Property Name	Value
DataCleansing Connector - Vendor1	"namefirm"
namefirm DataCleansing Field 1	"Last Name", "drl_ilname1", "drl_last_name_std1"
namefirm DataCleansing Field 2	"First Name", "drl_ifname1", "drl_first_name_std1"
namefirm DataCleansing Field 3	"Middle Name", "drl_imname1", "drl_mid_name_std1"
namefirm DataCleansing Field 4	"Job Title", "drl_ititle1", "drl_title_std1"

## Preconfigured List Mgmt Prospective Contact Business Component User Properties

Table 50 shows the data matching user properties for the List Mgmt Prospective Contact business component.

**CAUTION:** MP\_KEYFLD\_KEYMISC is used to store the Siebel ROW\_ID during the data matching process; do not configure this user property.

Table 50. Preconfigured Firstlogic Data Matching Properties for the List Mgmt Prospective Contact Business Component

User Property Name	Value
DeDuplication Connector - Vendor1	"match"
match Type	Prospect
match DeDuplication Field 1	"MATCH_FIELD_LASTNAME", "Last Name"
match DeDuplication Field 2	"MATCH_FIELD_FIRSTNAME", "First Name"
match DeDuplication Field 3	"MATCH_FIELD_MIDDLENAME", "Middle Name"
match DeDuplication Field 4	"MATCH_FIELD_FIRM", "Account"
match DeDuplication Field 5	"MATCH_FIELD_UNPADDRLINE", "Street Address"
match DeDuplication Field 6	"MATCH_FIELD_CITY", "City"
match DeDuplication Field 7	"MATCH_FIELD_STATE", "State"
match DeDuplication Field 8	"MATCH_FIELD_ZIP4", "Postal Code"
match DeDuplication Field 9	"MATCH_FIELD_COUNTRY", "Country"
match DeDuplication Field 10	"MATCH_FIELD_FIRMLOC", "Primary Account Location"
match DeDuplication Field 11	"MP_KEYFLD_KEYMISC", ""

Table 51 shows the data cleansing user properties for the List Mgmt Prospective Contact business component.

Table 51. Preconfigured Firstlogic Data Cleansing Properties for the List Mgmt Prospective Contact Business Component

User Property Name	Value
DataCleansing Connector - Vendor1	"namefirm, address"
namefirm DataCleansing Field 1	"Last Name", "drl_ilname1", "drl_last_name_std1"
namefirm DataCleansing Field 2	"First Name", "drl_ifname1", "drl_first_name_std1"
namefirm DataCleansing Field 3	"Middle Name", "drl_imname1", "drl_mid_name_std1"

Table 51. Preconfigured Firstlogic Data Cleansing Properties for the List Mgmt Prospective Contact Business Component

User Property Name	Value
namefirm DataCleansing Field 4	"Job Title", "drl_ititle1", "drl_title_std1"
namefirm DataCleansing Field 5	"Account", "drl_ifirmname1", "drl_firm_std1"
namefirm DataCleansing Field 6	"Primary Account Location", "drl_ifirmloc1", "drl_firm_loc_std1"
address Type	Prospect Address
address DataCleansing Field 1	"Street Address", "nad_line_address1", "nad_line_address1"
address DataCleansing Field 2	"City", "nad_line_locality1", "nad_line_locality1"
address DataCleansing Field 3	"State", "nad_line_region1"
address DataCleansing Field 4	"Postal Code", "nad_line_postcode", "nad_line_postcode"
address DataCleansing Field 5	"Country", "nad_line_country"

## Preconfigured Business Address Business Component User Properties

Table 52 shows the data cleansing user properties for the Business Address business component.

Table 52. Preconfigured Data Cleansing Properties for the Business Address Business Component

User Property Name	Value
DataCleansing Connector - Vendor1	"namefirm, address"
address Type	Business Address
address DataCleansing Field 1	"Street Address", "nad_line_multiline1", "nad_line_multiline1"
address DataCleansing Field 2	"City", "nad_line_locality1", "nad_line_locality1"
address DataCleansing Field 3	"State", "nad_line_region1"
address DataCleansing Field 4	"Postal Code", "nad_line_postcode", "nad_line_postcode"
address DataCleansing Field 5	"Country", "nad_line_country"
address DataCleansing Field 6	"Street Address 2", "nad_line_multiline2", "nad_line_multiline2"



# E

## Finding and Using Siebel Data Quality Information

This chapter discusses where to find information relevant to your use of the Siebel Data Quality (SDQ) products. It includes the following topics:

- [“Important Siebel Data Quality Resources” on page 145](#)
- [“SDQ Seed Data” on page 147](#)

### Important Siebel Data Quality Resources

SDQ is just one of many server processes available in Oracle’s Siebel Business Applications. Some of these server processes contribute to SDQ functionality and are partly documented in this book. For more information about these server processes, and other information relevant to SDQ, see the following subtopics.

#### Technical Documentation

Use the following Siebel documentation as additional references when using SDQ:

- *Using Siebel Tools* for information about how to modify standard Siebel objects and create new objects to meet your organization’s business requirements.
- *Fundamentals* for general information about merging records.
- Siebel Deployment Documentation Suite, including:
  - *Siebel Installation Guide* for the operating system you are using for details on how to install SDQ 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 Business Applications using Siebel Tools
  - *Siebel Developer’s Reference* for detailed descriptions of business components, user properties, and so on
  - *Deployment Planning Guide* to familiarize yourself with the basics of the underlying Siebel application architecture.
  - *System Monitoring and Diagnostics Guide for Siebel Business Applications*.
  - *Going Live with Siebel Business Applications* for information about how to migrate customizations from the development environment to the production environment.
  - *Security Guide for Siebel Business Applications* for information about built-in seed data in the enterprise database, such as employee, position, and organization records.

- *Performance Tuning Guide* for information about tuning and monitoring specific areas of the Siebel applications architecture and infrastructure, such as the object manager infrastructure.
- *Siebel Data Model Reference* for information about how data used by Siebel applications 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 Business Applications.
- *Siebel eScript Language Reference* for information about writing scripts to extend SDQ functionality.
- *Applications Administration Guide* for general information about administering Siebel Business Applications.
- *Upgrade Guide* or *Upgrade Guide for DB2 UDB for z/OS* for information about upgrading your installation.

## Third-Party Documentation

The following third-party documentation that is included on the *Siebel Business Applications Third-Party Bookshelf* must be used as additional references when using SDQ:

- SSA-NAME3 software documentation from Identity Systems (formerly known as Search Software America). This documentation provides information you need to configure and administer data matching using the SDQ Matching Server.
- Firstlogic software documentation. This documentation provides information you need to configure and administer data matching and data cleansing using Firstlogic.

## Siebel SupportWeb

The Siebel technical support Web site (SupportWeb) is located at <http://ebusiness.siebel.com/supportweb>. It provides search-engine access to the Siebel Bookshelf, Technical Notes, Siebel Alerts, troubleshooting information, and other important information:

- **System Requirements and Supported Platforms on Siebel SupportWeb.** A definitive list of system requirements and supported platforms for a release, including the following:
  - Information on supported third-party products
  - A description of supported upgrade paths
  - Lists of product and feature limitations; either unavailable in the release or in certain operating environments

This information is located on the SupportWeb Knowledge Base at Product Documentation > System Requirements and Supported Platforms & Miscellaneous Documentation.

- **Release Notes on Siebel SupportWeb.** The most current information on known product anomalies and workarounds and any late-breaking information not contained in this book. Release Notes are located on the SupportWeb Knowledge Base at Product Documentation > Release Notes.

- **Maintenance Release Guides.** Important information about updates to applications in maintenance releases. Maintenance release guides are located on SupportWeb at Product Documentation > Maintenance Release Guides.
- **Technical Notes.** Important information on specific SDQ issues.
- **FAQs.** Important information on various SDQ topics. FAQs of particular importance include:
  - FAQ 1593: Why are users able to create duplicate contact records in Siebel 7 application?
  - FAQ 1843: How can users enable Personal Address data cleansing with Firstlogic?
- **Siebel Alerts.** Time-critical information on key product behaviors and issues. Siebel Alerts about SDQ issues are located on SupportWeb at Siebel Alerts. One Siebel alert of particular importance is:
  - Alert 611: Merging Contact Records with Siebel Data Quality (SSA) May Not Work as Expected
- **Siebel Error Messages.** Important information about errors that have occurred in the Siebel Business Applications and additional information that can be used to correct the error conditions. Error messages related to SDQ are located on SupportWeb at Siebel Error Messages > Siebel eBusiness, Industry, Midmarket Edition, and CRM Applications Version 7.5.3 and later > Data Quality Manager - DQM.

**TIP:** Check SupportWeb regularly for updates.

## SDQ Seed Data

Siebel Business Applications include a sample database that contains example data of various kinds that you can use in demonstrating, evaluating, or experimenting with the Siebel client and Siebel Tools. While you can use the sample database to test real-time data matching, you cannot use it to test batch data matching, because that requires a running Siebel Server.

For more information about the sample database, see *Siebel Installation Guide* for the operating system you are using.

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 *Security Guide for Siebel Business Applications*.



# Index

## A

### architecture

- Siebel Data Quality (SDQ) Matching Server 15
- Siebel Data Quality (SDQ) Universal Connector 15

## B

**batch data cleansing, about** 48

**batch data matching, about** 48

### batch mode

- data cleansing, using batch mode 51
- data matching using the SDQ Universal Connector 53
- data quality component jobs, customizing 56
- described 12
- full data matching jobs 54
- generating keys, using batch mode 51
- incremental data matching jobs 55
- sample SDQ component customizations 56

### business components

- data cleansing, configuring to support 86
- data cleansing, process of configuring for 76
- data cleansing, troubleshooting 64
- data matching tip, about changing values 82, 85
- data matching, configuring to support 78, 83
- data matching, process of configuring for 76

### business service

- DataCleansing 15
- DeDuplication 15

### Business Service methods

- data cleansing scenario 103
- Get Siebel Fields 104
- Parse 104
- Parse All 105
- Value Match 100

## C

**candidate records** 21

### configuration options

- data cleansing, configuring business component to support 86
- data cleansing, process of configuring for 76
- data matching, configuring business components to support 78, 83

- data matching, process of configuring for SDQ Universal Connector, associating connector to a business component 77
- tip, about changing business component values 82, 85

### connector

See Siebel Data Quality (SDQ) Universal Connector

### connector mappings

- adding a field mapping 70
- external vendors, configuring for 68
- field mappings, about 67

## D

### data cleansing

- about 11
- Account business component user properties 107, 136, 141
- batch data cleansing, about 48
- batch job parameters 49
- batch mode 12
- batch mode, about running data cleansing 51
- business components, configuring to support 86
- business components, process of configuring for 76
- Business Service method scenario 103
- data quality component jobs for batch mode, customizing 56
- defined 11
- disabling for specific records 44
- disabling without restarting 40
- field mappings, about 67
- Get Siebel Fields method 104
- levels of enabling and disabling 33
- optimizing performance 93
- Parse All method 105
- Parse method 104
- Parse method invocation 105
- real-time mode 12
- real-time mode, about running in 47
- sample SDQ component customizations for batch mode 56
- SDQ Universal Connector 17
- troubleshooting 64

**data matching**

- about 11
- batch data matching, about 48
- batch job parameters 49
- batch mode 12, 53
- business components, configuring to support 78, 83
- business components, process of configuring for 76
- data quality component jobs for batch mode, customizing 56
- defined 11
- disabling without restarting 40
- duplicate records, filtering 61
- duplicate records, merging 61
- duplicate records, process of filtering and merging 60
- full data matching jobs 54
- Get Siebel Fields method invocation 104
- incremental data matching jobs 55
- levels of enabling and disabling 33
- optimizing performance 93
- real-time mode 12
- real-time mode, about running in 47
- real-time mode, enabling using command line 42
- sample SDQ component customizations for batch mode 56
- SDQ Matching Server 18
- SDQ Universal Connector 18
- sequenced merges, about 60
- sequenced merges, field characteristics 60
- Siebel Data Quality settings, applying 48
- troubleshooting 64
- user preference options, setting 42
- Value Match method input property sets 100
- Value Match method invoked from example 102
- Value Match method output property sets 101
- Value Match method scenario 99
- Value Match method, about 99, 100
- data model, SDQ Matching Server** 24
- data quality component jobs**
  - customizing 56
  - sample SDQ component customizations for batch mode 56
- data quality data cleansing field values, about using** 73
- Data Quality Manager**
  - about using 48
  - customized component jobs, creating 56
- Data Quality settings**
  - user preference options, setting 42

**data quality settings**

- applying 48
- Enable DataCleansing 38
- Enable DeDuplication 38
- Force User Dedupe Account 38
- Force User Dedupe Contact 38
- Force User Dedupe List Mgmt 38
- Fuzzy Query - Max Returned 39
- Fuzzy Query Enabled 39
- Key Type 20, 39
- Match Threshold 23, 39
- Search Type 22, 39
- specifying 37
- user preference options, setting 42
- data, seed** 147
- DataCleansing business service** 15
- DeDup Key Modification Date field**
  - using for batch generations 81
  - using for performance reasons 97
- Dedup Query Expression** 21
- Dedup Token Expression** 21
- DeDuplication business service** 15
- duplicate records** 23
- dynamic link libraries (DLLs)**
  - libraries supported 28, 31
  - vendor 113

**E**

- Enable Data Cleansing field** 43
- Enable DataCleansing data quality setting** 38
- Enable DeDuplication data quality setting** 38
- external callers, invoking SDQ from** 99

**F**

- features, new** 9
- fields, mapping**
  - deactivating 73
  - mapping Universal Connector data cleansing fields to Siebel business component fields 71
  - mapping Universal Connector data matching fields to Siebel business component fields 68
- filtering, duplicate records** 61
- Force User Dedupe Account data quality setting** 38
- Force User Dedupe Contact data quality setting** 38
- Force User Dedupe List Mgmt data quality setting** 38
- fuzzy query** 26

- configuring mandatory fields 89
- enabling and disabling 44
- example of enabling for use with
  - Accounts 63
- identifying mandatory fields 45
- using 62

### **Fuzzy Query - Max Returned data quality setting** 39

### **Fuzzy Query Enabled data quality setting** 39

## **G**

### **generating keys**

- about 19
- batch mode, about generating keys 51

### **Get Siebel Fields method** 104

- about 104
- invocation 104

## **I**

### **installing**

- SDQ Matching Server installation files 27
- Siebel Data Quality (SDQ) Universal Connector 30
- third-party software, about installing for using with SDQ Universal Connector 31

## **K**

### **key generation**

- about 19
- DeDup Key Modification Date field, using for batch generations 81

### **key refresh**

- about 19
- DeDup Key Modification Date field, using for batch generations 81

### **Key Type data quality setting** 39

### **keys**

- about match key generation 19
- Dedup Token Expression 19

## **L**

### **libraries, dynamic link libraries (DLLs), supported** 28, 31

## **M**

### **match key generation, about** 19

### **match key, defined** 19

### **match purpose**

- about 90
- configuration example 78
- configuring 91

### **match scores**

- calculation 22
- SDQ Matching Server 22
- SDQ Universal Connector 22

### **Match Threshold data quality setting** 23, 39

### **Merge button** 59

### **Merge Records option** 59

### **merging** 59

- duplicate records 59, 61
- duplicate records, process of merging 60
- sequenced merges, about 60
- sequenced merges, field characteristics 60

### **modes, operation modes described** 12

## **N**

### **new features** 9

## **P**

### **Parse All method** 105

### **Parse method** 104

- about 104
- invocation 105

### **performance**

- data cleansing, optimizing 93
- data matching, optimizing 93
- SDQ Matching Server, optimizing 94

### **pop-up windows**

- configuring for data matching 89

### **potential duplicates** 23

## **R**

### **real-time mode**

- data matching and data cleansing, about running 47
- described 12

### **records**

- data cleansing, disabling for specific records 44
- duplicate records, filtering 61
- duplicate records, merging 59, 61
- duplicate records, process of filtering and merging 60
- sequenced merges, about 60
- sequenced merges, field characteristics 60

## **S**

### **SDQ Matching Server**

- See Siebel Data Quality (SDQ) Matching Server

### **SDQ Universal Connector**

- See Siebel Data Quality (SDQ) Universal Connector

### **Search Type data quality setting** 22, 39

**searching**

- duplicate records, process of searching for 60

**seed data** 147**sequenced merges**

- about 60
- field characteristics 60

**Siebel Data Quality (SDQ)**

- configuration options 66
- data cleansing using batch jobs 51
- data matching using batch jobs 53
- generating keys using batch jobs 51
- modes of operation 12
- product modules 12
- sample SDQ component customizations for
  - batch mode 59
- troubleshooting 64

**Siebel Data Quality (SDQ) Matching Server**

- about 13
- architecture 15
- comparing to SDQ Universal Connector 14
- data matching 18
- data model 24
- dynamic link libraries (DLLs), supported 28
- identifying candidate records 21
- installation files 27
- mapping vendor fields to business component fields 68
- match key generation 20
- match scores 22
- optimizing performance 94
- sample SDQ component customizations for
  - batch mode 56
- SSA-NAME3 software 13
- upgrading from Siebel 7.7 30
- viewing preconfigured connector properties for 107
- viewing preconfigured field mappings for 108

**Siebel Data Quality (SDQ) Universal Connector**

- about 13
- architecture 15
- business component, associating the connector to 77
- comparing to SDQ Matching Server 14
- data cleansing 17
- data matching 18
- dynamic link libraries (DLLs), supported 31
- fields, mapping to Siebel business component fields. 71
- identifying candidate records 21
- installing, about 30
- mapping vendor fields to business component

- fields 69
- match key generation 19
- match scores 22
- new connectors, process of configuring 73
- new connectors, registering 74
- preconfigured connector properties 133
- preconfigured field mappings 135
- third-party software, about installing 31

**Siebel Data Quality resources** 145**Siebel Data Quality seed data** 147**Siebel Data Quality settings**

- applying 48
- user preference options, setting 42

**Siebel Data Quality software**

- enabling at the Enterprise level 35
- enabling at the Object Manager level 40

**Siebel Server**

- disabling data cleansing without restarting server 40
- disabling data matching without restarting server 40

**SSA-NAME3 software** 13**T****table size recommendations** 94**third-party software**

- SDQ Universal Connector, installing for use with 31
- SDQ Universal Connector, using with 13

**troubleshooting**

- data cleansing 64
- data matching 64

**U****Universal Connector API**

- batch mode data cleansing functions 129
- connector initialization and termination functions 114
- data cleansing and data matching algorithms 130
- error message functions 119
- parameter setting functions 116
- real-time data cleansing functions 128
- real-time data matching functions 119
- sdq\_close\_session function 116
- sdq\_data\_cleanse function 129
- sdq\_datacleanse function 128
- sdq\_dedup\_realtime function 119
- sdq\_get\_error\_message function 119
- sdq\_init\_connector function 114
- sdq\_init\_session function 115
- sdq\_set\_global\_parameter function 117
- sdq\_set\_parameter function 118



- sdq\_shutdown\_connector function 115
- session initialization and termination
  - functions 115
- upgrading the SDQ Matching Server from Siebel 7.7** 30
- user preference options, setting** 42
- user properties**
  - Account business component user properties 107, 136, 141
  - DataCleansing Connector 67
  - DeDuplication Connector 67
  - library base name 74

## V

### Value Match method

- data matching scenario 99
- input property sets 100
- invoked from example 102
- output property sets 101
- workflow, about invoking using 103

### vendors

- connector mappings, adding a field mapping 70
- connector mappings, configuring for external vendors 68
- data quality field mappings, about 67
- rules for dynamic link libraries 113

