Siebel Data Quality Administration Guide

Version 8.0, Rev. C

July 2011

Contents

Chapter 5: Enabling and Disabling Data Matching and Data Cleansing

Chapter 6: Configuring Siebel Data Quality

Chapter 7: Using Siebel Data Quality

Chapter 8: Optimizing Siebel Data Quality Performance

Appendix A: Setting Up Oracle Data Quality Matching Server for Data Matching

Appendix B: Examples of Parameter and Field Mapping Values for Universal Connector

Appendix C: Preconfigured Parameter and Field Mapping Values for SDQ Matching Server

Appendix D: Universal Connector API

Appendix E: Siebel Business Applications Action Sets

1 What's New in This Release

What's New in Siebel Data Quality Administration Guide, Version 8.0, Rev. C

This guide has been updated to remove third-party content that is no longer supported. Table 1 lists changes described in this version of the documentation to support release 8.0 of the software.

Table 1. New Product Features in Siebel Data Quality Administration Guide, Version 8.0, Rev. C

Topic	Description
"Universal Connector Parameter and Field Mapping Values for ODQ Address Validation Server" on page 139	New topic. Shows the preconfigured parameter and field mapping values for ODQ Address Validation Server.

What's New in Siebel Data Quality Administration Guide, Version 8.0, Rev. B

Table 2 lists changes described in this version of the documentation to support release 8.0 of the software.

Table 2. New Product Features in Siebel Data Quality Administration Guide, Version 8.0, Rev. B

Topic	Description
"SDQ Matching Server Libraries" on page 32	Updated topic. The language and population information for Siebel Data Quality Matching Server libraries has been updated.
"Configuring Real-Time Deduplication Window for Child Applets" on page 66	New topic. Describes how to configure the real-time DeDuplication Window for child applets.
"Siebel Data Quality User Properties" on page 69	New appendix. Provides detailed information about deduplication and data cleansing user properties. This information was formerly in *Siebel Developer's Reference*.
"Merge Algorithm in the Object Manager Layer" on page 89	New topic. Provides detailed information about how the merge records algorithm works.
"Installing ODQ Matching Server on UNIX" on page 117	New topic. Describes how to install Oracle Data Quality (ODQ) Matching Server on a UNIX operating system.
"Configuring ODQ Matching Server on UNIX" on page 122	New topic. Describes how to configure ODQ Matching Server on a UNIX operating system.
"Universal Connector Parameter and Field Mapping Values for ODQ Matching Server" on page 136	New topic. Describes the Universal Connector parameter and field mapping values for the ODQ Matching Server.

Additional Changes

Version 8.0, Rev. B also contains the following changes:

- The product name, Identity Search Server (ISS), has changed to Informatica Identity Resolution (IIR).
- The product name, ODQ Cleansing Server, has changed to ODQ Address Validation Server.

What’s New in Siebel Data Quality Administration Guide, Version 8.0, Rev. A

Table 3 lists changes described in this version of the documentation to support release 8.0 of the software.

Table 3. New Product Features in Siebel Data Quality Administration Guide, Version 8.0, Rev. A

Topic	Description
Appendix A, “Setting Up Oracle Data Quality Matching Server for Data Matching.”	New appendix. Describes how to set up and configure ODQ Matching Server for data matching.

What’s New in Siebel Data Quality Administration Guide, Version 8.0

Table 4 lists changes described in this version of the documentation to support release 8.0 of the software.

Table 4. New Product Features in Siebel Data Quality Administration Guide, Version 8.0

Topic	Description
Deduplication on address update. See “Data Matching” on page 20.	When the primary address data of an account, contact, or prospect record is updated, the match keys are regenerated as follows: ■ Keys are regenerated for an account only when a primary address is selected for the account, or the primary address is changed for the account. ■ If an account already has an address and the user modifies part of the address record (for example, city), then the keys are not regenerated. Previously match keys were not updated, and no deduplication was triggered.
Multiple key query support. See “Match Key Generation with the Universal Connector” on page 22.	For the Universal Connector, multiple keys are now generated for a single record. Previously only one match key was generated for each record.

Table 4. New Product Features in Siebel Data Quality Administration Guide, Version 8.0

Topic	Description
Data quality rules. See "Data Quality Rules" on page 87 and "Creating a Data Quality Rule" on page 87.	The Administration - Data Quality screen, Rules view allows the administrator to set up rules for each data quality operation. The data quality rules specify the parameters used when a data quality operation is performed in real-time or in batch mode.
Third-party software vendor administration. See "Configuring Vendor Parameters" on page 57, "Examples of Parameter and Field Mapping Values for Universal Connector" on page 135, and Appendix C, "Preconfigured Parameter and Field Mapping Values for SDQ Matching Server."	The information about data quality vendors is now administered in the Administration - Data Quality screen, Third Party Administration view rather than by using Siebel Tools. You can therefore make configuration changes without having to recompile the Siebel repository. Vendor-specific parameters are configured in the Vendor Parameter view. Vendor field mappings are configured in a BC Vendor Field Mapping view for each business component for which data cleansing or data matching is supported.

Additional Changes

Version 8.0 also contains the following changes:

- The tables of preconfigured parameters for the SDQ Matching Server now contain descriptions of each parameter; see "Preconfigured Vendor Parameters for SSA" on page 144.
- Changes to improve the organization of the information.

2 Overview of Siebel Data Quality

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

Data Cleansing and Data Matching

The data stored in account, contact, and prospect 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 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.

Data Quality Product Modules

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

- **Siebel Data Quality (SDQ) Matching Server.** Provides real-time and batch data matching functionality using embedded SSA-Name3 software from Identity Systems (formerly Search Software America).
- **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. For more information, see "SDQ Universal Connector."

  NOTE: The SDQ Universal Connector is currently used by some partners.

- **Oracle Data Quality (ODQ) Matching Server.** This is a newly released product providing real-time and batch data matching functionality using licensed third-party Informatica Identity Resolution (IIR) software. For more information, see "Setting Up Oracle Data Quality Matching Server for Data Matching" on page 111.
- **Oracle Data Quality (ODQ) Address Validation Server.** This is a newly released product providing an address validation and standardization tool covering more than 240 countries.

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 CRM transactional database.

The Matching Server uses embedded SSA-NAME3 software from Identity Systems. The SSA-NAME3 DLLs (Windows) or shared libraries (UNIX) are embedded in Siebel Business Applications and are installed with Siebel Software Installers for Windows and UNIX operating systems. The Matching Server does not require additional third-party software installations to function.

The Matching Server works across the languages and operating systems supported by Siebel Business Applications. There are different DLLs or shared libraries 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 "SDQ Matching Server Libraries" on page 32.
- Platforms supported, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.
- SSA-NAME3 software, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.

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.

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 Oracle. For information about third-party solutions and about products that are certified for the Universal Connector, see the the Alliances section and the Partners section at

http://www.oracle.com/siebel

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 operating systems, though the support offered by particular third-party software for data matching or data cleansing might not cover all of the languages supported by Siebel Business Applications.

For more information about:

- Platforms supported, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.
- Third-party software, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.

SDQ Product Module Comparison

Table 5 provides a summary and comparison of the capabilities of the two SDQ product modules—the Matching Server and the Universal Connector.

Table 5. Siebel Data Quality Product Module Comparison

Topic	Matching Server	Universal Connector
Provides data matching for account, contact, and prospect data within Siebel Business Applications	Yes	Yes
Provides data cleansing for account, contact, prospect and business address data within Siebel Business Applications	No	Yes
Identifies duplicate records stored in accounts, contacts, and prospects data table	Yes	Yes
Supports multiple languages and operating systems	Yes	Yes

Table 5. Siebel Data Quality Product Module Comparison

Topic	Matching Server	Universal Connector
Calls SDQ functionality through standard Siebel CRM business services	Yes	Yes
Requires additional third-party software	No	Yes
Runs in real-time and batch modes	Yes	Yes

ODQ Matching Server

Oracle Data Quality (ODQ) Matching Server provides real-time and batch data matching functionality using licensed third-party Informatica Identity Resolution (IIR) software Version 2.7.

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

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

About Using the ODQ Matching Server

You can use the ODQ Matching Server to do the following:

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

For more information about ODQ Matching Server installation and configuration, see Appendix A, “Setting Up Oracle Data Quality Matching Server for Data Matching.”

ODQ Address Validation Server

The Oracle Data Quality (ODQ) Address Validation Server provides capabilities to parse, standardize, transliterate, duplicate, and validate all address data, resulting in improved address data quality.

The ODQ Address Validation Server provides:

- Quick validation and correction of worldwide postal addresses.
- Lower direct marketing and CRM costs.
- Address coverage for more than 240 countries.

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.

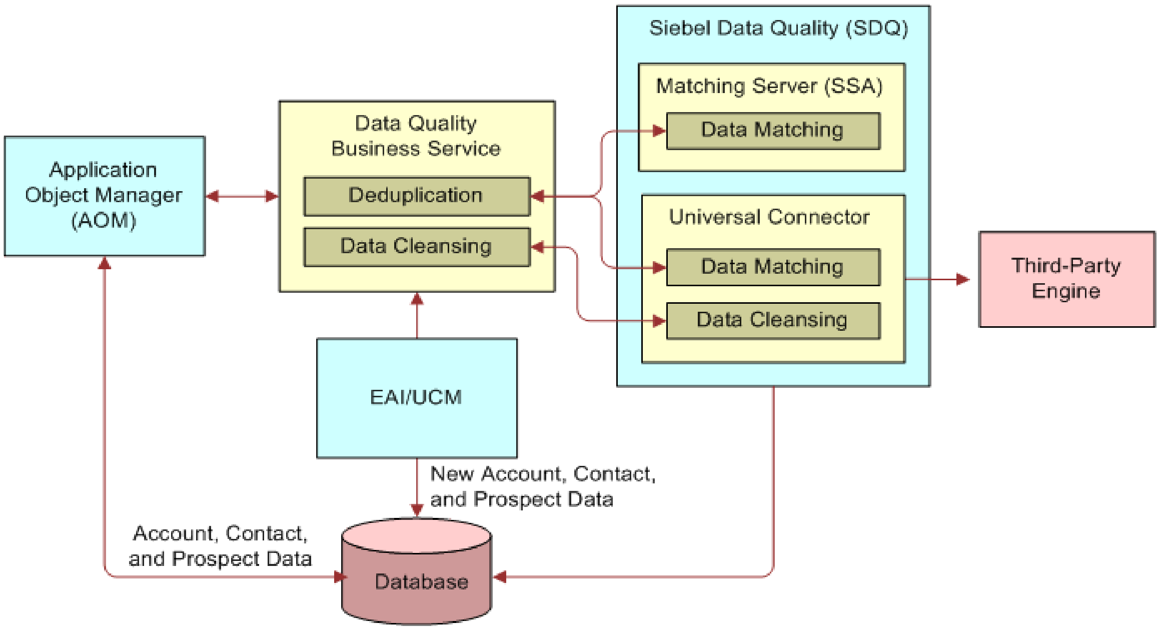
Siebel Data Quality (SDQ)
Matching Server (SSA)
Data Matching
Universal Connector
Data Matching
Data Cleansing
Data Quality Business Service
Deduplication
Data Cleansing
Application Object Manager (AOM)
Third-Party Engine
EAI/UCM
New Account, Contact, and Prospect Data
Account, Contact, and Prospect Data
Database

Figure 1. Siebel Data Quality Architecture

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

In batch mode, the Matching Server and Universal Connector are called by the preconfigured server component, Data Quality Manager (DQMgr), either from the Siebel application user interface, or by starting tasks with the Siebel Server Manager command-line interface, the srvrmgr program. For more information, see *Siebel System Administration Guide*.

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 use the Universal Connector with third-party software for data cleansing on the same Siebel application object manager.

The Universal Connector and Matching Server obtain account, contact, and prospect field data from the Siebel CRM 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 or shared library, and in the case of the Universal Connector, this is a third-party software library depending on your configuration. 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 21.

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*.

NOTE: The *Siebel Bookshelf* is available on Oracle Technology Network (OTN) and Oracle E-Delivery. It might also be installed locally on your intranet or on a network location.

3 Siebel Data Quality Concepts

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

Data Cleansing

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

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

For each type of record, data cleansing is performed for the fields that are specified in the Third Party Administration view. The mapping between the Siebel application field names and the vendor field names is defined for each business component. For information about the preconfigured field mappings, see Appendix B, “Examples of Parameter and Field Mapping Values for Universal Connector.”

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 CRM 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 CRM database with the cleansed information and displays the cleansed information to the user.

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

If both data cleansing and data matching are enabled, data cleansing is done first. For information about running data cleansing batch jobs, see "Cleansing Data Using Batch Jobs" on page 78.

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 the current record against all other records of the same type, and with the same match keys, in the application using the fields specified in the Third Party Administration view. The mapping between the Siebel application field names and the vendor field names is defined for each business component. For information about the preconfigured field mappings for SSA, see "Preconfigured Field Mappings for SSA" on page 149.

SDQ performs matching using fields, for example, addresses, that can have multi-value group (MVG) values associated with the type of record being matched. However, SDQ is not currently able to match using 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 80.

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

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

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

3 The duplicate records are displayed either in a 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 following 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.

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

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 79.

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

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

For example, if there are records as follows:

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

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

The batch capability is useful in the following circumstances:

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

For instructions about using batch jobs to generate or refresh keys, see "Generating or Refreshing Keys Using Batch Jobs" on page 79.

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

The value of the match keys depend on a business component-specific Dedup Token Expression parameter, as shown in Table 6 on page 24.

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

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

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

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

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

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

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.

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, as follows:

- **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 plus John plus Alexander
  - Smith plus Alexander plus John
  - John plus Alexander plus Smith
  - John plus Smith plus Alexander
  - Alexander plus John plus Smith

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

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

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

See Table 6 for information about how the default expressions differ for different business components.

Table 6. Expressions Used for Keys and Queries (Example)

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

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

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

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, as follows:

- **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. The software calculates a match score from 0 to 100 to indicate the degree of similarity between the candidate records and the current record.

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

The algorithms used to calculate match scores are complex. These algorithms are the intellectual property of third party software vendors; Siebel Business Applications 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. For help with tailored matching rules, create a service request (SR) on My Oracle Support. Alternatively, you can phone Oracle Global Customer Support directly to create a service request or get a status update on your current SR. Support numbers are listed on My Oracle Support.

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 Windows Displayed in Real-Time Data Matching" on page 64.

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 Matching and Cleansing" on page 74.

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 91.

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

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

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

Data Model for the SDQ Matching Server

This topic provides information about the database tables that the SDQ Matching Server interacts with. Table 7 lists the tables that are relevant to data matching. Table 8 on page 28 lists the database operations that result from the various data matching functions.

Table 7. Tables Used by SDQ Matching Server

Siebel CRM 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: ■ **DUP_OBJ_ID field.** Stores the row ID of a potential duplicate record. ■ **OBJ_ID field.** Stores the row ID of the master record (the record for which matching was performed). ■ **OBJ_NAME field.** Stores the name of the business component for which data matching was performed, for example, Account. ■ **TOT_SCORE_VAL field.** Stores the match score of the potential duplicate record.
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 8. Database Operations Resulting from SDQ Matching Functions

Matching Function	Database Operation
Key Generation or Key Refresh	■ Data creation in one of the following tables: ■ S_ORG_DEDUP_KEY (Account records) ■ S_PER_DEDUP_KEY (Contact records) ■ S_PRSP_DEDUPKEY (List Mgmt Prospective Contact records) ■ Data update in one of the following tables: ■ S_CONTACT (Contact records) ■ S_ORG_EXT (Account records) ■ S_PRSP_CONTACT (List Mgmt Prospective Contact records) The DEDUP_KEY_UPD_DT column is updated.
Matching (DeDuplication)	■ Data lookup in one of the following tables: ■ S_ORG_DEDUP_KEY (Account records) ■ S_PER_DEDUP_KEY (Contact records) ■ S_PRSP_DEDUPKEY (List Mgmt Prospective Contact records) ■ Data creation in the S_DEDUP_RESULT table ■ Data update in one of the following tables: ■ S_CONTACT (Contact records) ■ S_ORG_EXT (Account records) ■ S_PRSP_CONTACT (List Mgmt Prospective Contact records) The DEDUP_LAST_MTCH_DT column is updated.
Merging	■ Data deletion in the S_DEDUP_RESULT table ■ Data update and deletion in one of the following tables: ■ S_CONTACT (Contact records) ■ S_ORG_EXT (Account records) ■ S_PRSP_CONTACT (List Mgmt Prospective Contact records)

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 the * (asterisk) character. 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 24.
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 engine to evaluate the possible matches. The query result returns the best available matches, up to the number of records specified by Fuzzy Query Max Results.
   - If the preliminary query results contain fewer records than the value of the Fuzzy Query Max Results setting, then 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 51.

For information about using fuzzy query, see "Using Fuzzy Query" on page 94.

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

As a preliminary step in installing SDQ software, including the SDQ Matching Server software, you must use the Siebel Image Creator utility and Siebel CRM media files (from your DVD or FTP site) to create a network-based Siebel CRM 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 "SDQ Matching Server Libraries" on page 32.

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, and then Configure Siebel Server to start the utility manually.

Table 9 describes the SDQ Matching Server files and folders that are installed.

Table 9. Siebel Data Quality Matching Server Installation Files

Installation Component	Installation Information
SSA-NAME3 library files For more information about library files, see "SDQ Matching Server Libraries" on page 32.	For Windows: `SIEBSRVR_ROOT\bin\language_code\n3sgsb.dll` where *language_code* is the appropriate language code, such as ENU for U.S. English.
	For Oracle Solaris and AIX: `SIEBSRVR_ROOT/lib/language_code/n3sgsb.so` where *language_code* is the appropriate language code, such as ENU for U.S. English.
	For HP-UX: `SIEBSRVR_ROOT/lib/language_code/n3sgsb.sl` where *language_code* is the appropriate language code, such as ENU for U.S. English.

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 shared libraries adapted for different languages or language families. All of these shared libraries have the same name (for example, n3sqsb.dll on Windows), but are installed in language-specific subdirectories as shown in Table 9.

Each interactive object manager uses the language-specific library from the `\bin\<language>` folder on Windows or the `/lib/<language>` folder on UNIX respectively, and the keys that are generated have LANG_ALGRTHM_CD in the key table, which reflects the library's population and code page. Only records with the same LANG_ALGRTHM_CD values are considered for matching against each other.

The library in the ENU folder is the most generic library as it uses the `Default` (=`International`) population, which can be used to deduplicate records in all Latin languages. Latin languages are languages predominant in the Americas, Western Europe, Australia, and New Zealand.

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. Examples include GmBH (German), Oys (Finnish), and other abbreviations for corporate structures.

In addition, the Siebel CRM installation media includes matching libraries for other languages and code pages. You can retrieve these additional shared libraries by installing the other language packs on the Siebel Server. Table 10 on page 33 lists the languages supported.

For real-time matching, the object manager always uses the n3sqsb of its language. However, it is different for batch tasks. For batch tasks, the DQMgr by default also uses Language ENU with its international population library.

To use a different population or matching library (other than ENU) for batch deduplication, you must clone the DQMgr component and set its language parameter to the language of the library that you want to use. This is optional for Latin languages. For example, the DEU, FRA, and ITA libraries all result in slightly better matches when you clone the DQMgr (instead of using the international ENU library), but there is the added cost of having to create a separate DQMgr for each language and run separate batch tasks with the object WHERE clause set to process only records in that language.

If you are only using Western languages for real-time AND batch deduplication, you can copy the ENU n3sqsb to the other WESTERN (Latin) languages in the `\bin\<language>` folder (on Windows) or the `/lib/<language>` folder (on UNIX), so that all keys generated from real-time and batch data matching will have the same LANG_ALGRTHM_CD value of DefaultLatin_1_Mixed.

For non-latin languages (ARA, JPN, KOR, or THA), it is essential to create a separate DQMgr with the parameter language set accordingly in the component definition (as the library is loaded on first access and the language cannot be specified dynamically for batch tasks). For example:

- Create DQMgr_ARA where Language is equal to ARA (Language=ARA)

  NOTE: By default, the Application Repository File parameter changes to siebel.srf and you must change this if using a Siebel SIA application.

- Then run the batch tasks (Key Generate and DeDuplication) using an object WHERE clause setting that only retrieves records with Arabic data, for example, using the [Country] or [Language Code] fields.

NOTE: You must ensure that any fields that you use in the object WHERE clause or a rule's search spec are always populated through configuration in Siebel Tools, for example, by setting a predefault value and/or exposing the fields in the GUI and making them required.

Table 10. SDQ Matching Libraries and Supported Languages

Matching Library and Code Page	Languages Supported and Language Code	Syntax (Name and Value Setting)
International (“Default”) Latin_1_Mixed code page	DAN - Danish	SSA Population-Codepage DAN “Denmark”, “Latin_1_Mixed”
	DEU - German	SSA Population-Codepage DEU “Germany”, “Latin_1_Mixed”
	ENU - U.S. English	SSA Population-Codepage ENU “Default”, “Latin_1_Mixed”
	ESN - Spanish	SSA Population-Codepage ESN “Spain”, “Latin_1_Mixed”
	FIN - Finnish	SSA Population-Codepage FIN “Finland”, “Latin_1_Mixed”
	FRA - French	SSA Population-Codepage FRA “French”, “Latin_1_Mixed”
	ITA - Italian	SSA Population-Codepage ITA “Italy”, “Latin_1_Mixed”

Table 10. SDQ Matching Libraries and Supported Languages

Matching Library and Code Page	Languages Supported and Language Code	Syntax (Name and Value Setting)
International (“Default”) Latin_1_Mixed code page (continued)	NLD - Dutch	SSA Population-Codepage NLD “Netherlands”, “Latin_1_Mixed”
	PTB - Brazilian Portuguese	SSA Population-Codepage PTB = “Brazil”, “Latin_1_Mixed”
	PTG - Portuguese	SSA Population-Codepage PTG = “Portugal”, “Latin_1_Mixed”
	SVE - Swedish	SSA Population-Codepage SVE = “Sweden”, “Latin_1_Mixed”
NOTE: The following non-ENU n3sqsb libraries are single language libraries that will only match within the Population and Codepage specified.		
Arabic	ARA - Arabic	SSA Population-Codepage ARA “Arabic”, “Arabic”
Chinese (Simplified)	CHS - Simplified Chinese	SSA Population-Codepage CHS “China”, “Chinese_Simp”
Chinese (Traditional)	CHT - Traditional Chinese	SSA Population-Codepage CHT “China”, “Chinese_Trad”
Czech	CSY - Czech	SSA Population-Codepage CSY “Czech”, “Latin_2_1250”
Greek	ELL - Greek	SSA Population-Codepage ELL “Greece”, “Greek”
Hebrew	HEB - Hebrew	SSA Population-Codepage HEB “Israel”, “Hebrew”
Japanese	JPN - Japanese	SSA Population-Codepage JPN “Japan”, “Japanese”
Korean	KOR - Korean	SSA Population-Codepage KOR “South_Korea”, “Korean”
Polish	PLK - Polish	SSA Population-Codepage PLK “Poland”, “Latin_2_1250”
Thailand	THA - Thai	SSA Population-Codepage THA “Thailand”, “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 Siebel CRM Version 7.7

For CRM Version 7.8 or later, 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 Siebel CRM Version 7.7:

- Siebel CRM Version 7.8 and 8.0 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 79.

  Before you regenerate keys, determine whether you need different Data Quality Settings, for example, higher Match Threshold values with the new libraries.
- Match results might 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 Siebel CRM Version 7.8 and later 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 to regenerate all match keys, and for 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 CRM media files (from your DVD or FTP site) to create a network-based Siebel CRM 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 36.

Installing Third-Party Software for Use with the Universal Connector

This topic briefly 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.

SDQ Universal Connector Libraries

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

The names of the shared libraries are vendor-specific, but must follow naming conventions as described in "Vendor Libraries" on page 153.

The Siebel CRM installation process copies these DLL or shared library files to a location that depends on the operating system you are using, as shown in Table 11.

Table 11. Storage Locations for SDQ Universal Connector Library Files by Operating System

Does Vendor Support Multiple Languages?	DLL Storage Locations (Windows)	Shared Library Storage Locations (UNIX)

No	For Siebel Server:
	`Siebel_Server_root\bin\`
	For Developer Web Client:
	`Client_root\bin\`
		For Siebel Server:
		`Siebel_Server_root/lib`

Table 11. Storage Locations for SDQ Universal Connector Library Files by Operating System

Does Vendor Support Multiple Languages?	DLL Storage Locations (Windows)	Shared Library Storage Locations (UNIX)
Yes	For Siebel Server: *Siebel_Server_root*\bin\language_code For Developer Web Client: *Client_root*\bin\language_code where *language_code* is the appropriate language code, such as ENU for U.S. English.	For Siebel Server: *Siebel_Server_root*/lib/*language_code*

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

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

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

In the Siebel application there are various levels at which you can enable or disable data cleansing and data matching as summarized in Table 12. 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 12. Levels of Enabling and Disabling Data Matching and Cleansing

Screen and View	Setting or Parameter	Value	Parameters Set By
Values set at the enterprise level			
Administration - Server Configuration, Enterprises, Data Quality view	Enable button Disable button		Application administrator
Administration - Server Configuration, Enterprises, Parameters view	DeDuplication Data Type Data Cleansing Type	*Data Matching Vendor Name* *Data Cleansing Vendor Name*	

Table 12. Levels of Enabling and Disabling Data Matching and Cleansing

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

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

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

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

For more information, see the following topics:

- "Enabling Siebel Data Quality at the Enterprise Level" on page 41
- "Specifying Data Quality Settings" on page 43
- "Enabling Siebel Data Quality at the Object Manager Level" on page 46
- "Enabling Siebel Data Quality at the User Level" on page 49

Enabling Siebel Data Quality at the Enterprise Level

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

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

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

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

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

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

To enable SDQ at the enterprise level

1. Log in to the Siebel application with administrator responsibilities.
2. Navigate to the Administration - Server Configuration screen, then the 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.

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

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

1. Log in to the Siebel application with administrator responsibilities.
2. Navigate to the Administration - Server Configuration screen, then the 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.
   - **ISS.** Indicates that the ODQ Matching Server is used for data matching.

     NOTE: For deduplication with ODQ Matching Server to be active, you must change the DeDuplication Data Type from SSA to the *name of the third-party server* on all object managers and server components.

   - **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, then the 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 13. The parameters applicable only to the Matching Server are described in Table 14 on page 45

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 13. Data Quality Settings Applicable to Data Quality Product Module

Parameter	Description
Enable DataCleansing	Determines whether real-time data cleansing is enabled for the Siebel Server the administrator is currently logged into. The default value is Yes. Other values you set for SDQ can override this setting. For more information about this, see "Levels of Enabling and Disabling Data Cleansing and Data Matching" on page 39.
Enable DeDuplication	Determines whether real-time data matching is enabled for the Siebel Server the administrator is currently logged into. The default value is Yes. Other values you set for SDQ can override this setting. For more information about this, see "Levels of Enabling and Disabling Data Cleansing and Data Matching" on page 39.
Force User Dedupe - Account	Determines whether duplicate records are displayed in a window when a user saves a new account record. The user can then merge duplicates. If set to No, duplicates are not displayed in a window, but the user can merge duplicates in the Duplicate Accounts view. The default value is Yes. For more information about pop-up window, see "Configuring the Windows Displayed in Real-Time Data Matching" on page 64.

Table 13. Data Quality Settings Applicable to Data Quality Product Module

Parameter	Description
Force User DeDupe - Contact	Determines whether duplicate records are displayed in a window when a user saves a new contact record. The user can then merge duplicates. If set to No, duplicates are not displayed in a window, but the user can merge duplicates in the Duplicate Contacts view. The default value is Yes. For more information about window configuration, see "Configuring the Windows Displayed in Real-Time Data Matching" on page 64.
Force User DeDupe - List Mgmt	Determines whether duplicate records are displayed in a window when a user saves a new prospect record. The user can then merge duplicates. If set to No, duplicates are not displayed in a window, but the user can merge duplicates in the Duplicate Prospects view. The default value is Yes. For more information about window configuration, see "Configuring the Windows Displayed in Real-Time Data Matching" on page 64.
Fuzzy Query Enabled	Determines whether fuzzy query, an advanced search feature, is enabled. The default value is No. For more information about fuzzy query, see "Enabling and Disabling Fuzzy Query" on page 51
Fuzzy Query - Max Returned	Specifies the maximum number of records returned when a fuzzy query is performed. The default value is 500. For more information about fuzzy query, see "Enabling and Disabling Fuzzy Query" on page 51.
Match Threshold	Specifies a threshold above which any record with a match score is considered a match. Higher scores indicate closer matches (a perfect match is equal to 100). Possible values are: 50-100.

Table 14. Data Quality Settings for the SDQ Matching Server Only

Parameter	Description
Key Type	Determines the number of match keys generated. Possible values are: ■ **Limited.** The key generation algorithm performs only the most common permutations and generates fewer keys. ■ **Standard.** The key generation algorithm performs a wider set of permutations to provide the most exhaustive range of keys. 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 is equal to 100). Possible values are: 50-100.
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 called when a new or modified record is saved. Real-time data matching and cleansing is supported only for employee-facing applications. By specifying data matching and cleansing parameters at the object manager level in the Siebel application, you can enable data matching or cleansing for one application and disable it for another application. However, you cannot enable data matching for both the Matching Server and the Universal Connector for the same application.

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

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

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

- "Enabling SDQ Using the GUI" on page 46
- "Enabling SDQ Using the Command-Line Interface" on page 48

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 41.

Enabling SDQ Using the GUI

To enable SDQ at the Object Manager level using the GUI, complete the steps in the following procedure.

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, then the Servers view.
3. In the Components list, select an object manager where end users enter and modify customer data.

   For example, select the Call Center Object Manager (ENU) if you want to enable or disable real-time data matching or cleansing for that object manager.
4. Click the Parameters subview tab.
5. In the Parameters field in the Component Parameters list, apply the appropriate settings to the parameters 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 srvrmgr or by completing the following sub-steps:

   a Navigate to the Administration - Server Management screen, then the Servers view.

   b Click the Components Groups view tab (if not already active).

   c In the Servers list (upper applet), select the appropriate Siebel Server (if you have more than one in your enterprise).

   d In the Components Groups list (middle applet), select the component of your object manager, and use the Startup and Shutdown buttons to restart the component.

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

Enabling SDQ Using the Command-Line Interface

To enable SDQ at the Object Manager level using the command-line interface, complete the steps in the following procedure.

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 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=ISS for component
     SCCObjMgr_enu
     ```

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

     ```
     change parameter DataCleansingEnable=True, DataCleansingType=Vendor1 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 screen. However, a choice to disable a feature in the user preference settings takes priority (for the current user) over a choice to enable it in the Data Quality Settings view. The reverse is not true: if a feature is disabled in the Data Quality Settings view, you cannot override that disabling by enabling the feature in the user preferences settings.

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

To enable data quality at the user level

1 Log in to the SDQ application as the user.

2 Navigate to the User Preferences screen, then the Data Quality view.

3 In the Data Quality form, set the parameters for that user.

The following table describes the fields.

Field	Comments
Enable Data Cleansing	Select Yes to enable data cleansing for the current user. Otherwise, select No to disable data cleansing.
Enable DeDuplication	Select Yes to enable data matching for the current user. Otherwise select No to disable data matching.
Fuzzy Query Enabled	Select Yes to use a fuzzy query for the current user. Fuzzy query only works if certain conditions are met; see "Enabling and Disabling Fuzzy Query" on page 51. Select No to disable fuzzy queries for the current user.
Fuzzy Query - Max Matches Returned	Specify the maximum number of query result records you want SDQ to return to you. Valid values are 10 to 500. The default value is 100.
Key Type	Applicable only for the SDQ Matching Server. Select Limited to generate a smaller number of keys, or select Standard to generate a larger number of keys. For more information about Key Type settings, see Table 14 on page 45.

Field	Comments
Match Threshold	Applicable for the SDQ Matching Server and Universal Connector. Select a threshold above which any record with a match score is considered a match. Higher scores indicate closer matches (a perfect match is equal to 100). Possible values are: 50-100. If no threshold value is supplied in any of the data quality settings, the default value of 50 is used by the Siebel application.
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.

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

   There is no 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 29.

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 43.
- 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 49.
- 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 52.
- The query must leave optional fields blank.

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

Enabling Fuzzy Query

Use the following procedure to enable fuzzy query.

To enable fuzzy query

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

Disabling Fuzzy Query

Use the following procedure to disable fuzzy query.

To disable fuzzy query

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

For more information about fuzzy query, see "Using Fuzzy Query" on page 94, and "Example of Enabling and Using Fuzzy Query with Accounts" on page 95.

Identifying Mandatory Fields for Fuzzy Query

You might want to provide users with information about mandatory fields (query fields that must include values for the Siebel application to use fuzzy query mode). Table 15 shows the preconfigured mandatory fields that Siebel Business Applications provide.

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

Table 15. 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 must repeat Step 2 of this procedure.

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

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

6 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 53
- Process of Configuring New SDQ Connectors for the Universal Connector on page 54
- Configuring Vendor Parameters on page 57
- Mapping of Vendor Fields to Business Component Fields on page 57
- Configuring Business Components for Data Matching Using the Matching Server on page 60
- Configuring the Windows Displayed in Real-Time Data Matching on page 64
- Configuring the Mandatory Fields for Fuzzy Query on page 66
- Match Purpose on page 67
- Siebel Data Quality User Properties on page 69

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

Siebel Data Quality Configuration Overview

Table 16 summarizes the configuration that you can perform for SDQ.

Table 16. Siebel Data Quality Configuration Options

Configuration	See...
Configure vendor parameters. You can configure the parameters for each of the software vendors used by SDQ.	"Configuring Vendor Parameters" on page 57
Configure field mappings for business components. You can change or add field mappings.	"Mapping Data Matching Vendor Fields to Siebel Business Components" on page 58 and "Mapping Data Cleansing Vendor Fields to Siebel Business Component Fields" on page 59
Configure new connectors for data matching and data cleansing for the Universal Connector.	"Configuring Business Components for Data Matching Using the Matching Server" on page 60

Table 16. Siebel Data Quality Configuration Options

Configuration	See...
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.	"Configuring Business Components and Applets for Data Matching and Data Cleansing" on page 55
Configure the pop-up windows displayed in real-time data matching.	"Configuring the Windows Displayed in Real-Time Data Matching" on page 64
Configure the mandatory fields for fuzzy search.	"Configuring the Mandatory Fields for Fuzzy Query" on page 66
Configure SSA match purpose.	"Match Purpose" on page 67 and "Configuring Match Purpose" on page 68

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. To configure new SDQ connectors, perform the following tasks:

1. "Registering New SDQ Connectors" on page 54
2. "Configuring Business Components and Applets for Data Matching and Data Cleansing" on page 55

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

Registering New SDQ Connectors

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

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

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

The DLL Name setting in the Third Party Administration view tells the Siebel application how to load the DLL or shared library. The names of the libraries are vendor-specific, but must follow naming conventions as described in "Vendor Libraries" on page 153.

The Siebel application loads the libraries from the locations described in Table 11 on page 36.

To register a data quality connector

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

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

Configuring Business Components and Applets for Data Matching and Data Cleansing

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

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

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

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

To configure business components for data matching and data cleansing, complete the steps in the following procedure.

To configure business components for data matching and cleansing

1 Associate the business component with a connector.

This includes configuring the vendor parameters listed in the following table.

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

2 Configure the field mappings for each business component and operation.

3 Create a DeDuplication Results business component and add it to the Deduplication business object.

4 Configure an applet as the DeDuplication Results List Applet.

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

6 Add the business component user properties listed in the following table.

Property	Value
DeDuplication Results BusComp	The buscomp that you created in Step 3.
DeDuplication Results List Applet	The applet that you created in Step 4.

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

Configuring Vendor Parameters

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

There are preconfigured vendor parameters for the Matching Server with the embedded SSA software, and for the Universal Connector with ODQ Matching Server as an example.

For more information, see Appendix C, "Preconfigured Parameter and Field Mapping Values for SDQ Matching Server" and Appendix B, "Examples of Parameter and Field Mapping Values for Universal Connector."

To configure vendor parameters

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

Mapping of Vendor Fields to Business Component Fields

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

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

There are mappings for each supported business component and data quality operation (DeDuplication and Data Cleansing). There are preconfigured field mappings for SSA, see "Preconfigured Field Mappings for SSA" on page 149.

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

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

For more information about mapping vendor fields to business component fields, see the following:

- "Mapping Data Matching Vendor Fields to Siebel Business Components" on page 58
- "Adding a Field Mapping for Data Matching" on page 58
- "Mapping Data Cleansing Vendor Fields to Siebel Business Component Fields" on page 59

Mapping Data Matching Vendor Fields to Siebel Business Components

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

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

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

Adding a Field Mapping for Data Matching

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

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

The following procedure describes how to add a field mapping.

To add a field mapping for data matching

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

   The field mappings are displayed in the Field Mapping list.

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

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

6 If required, modify the corresponding real-time and batch mode data flows to incorporate the new field so that 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 Vendor Fields to Siebel Business Component Fields

Data cleansing is triggered when a record is saved after a field that is defined as an active data cleansing field is updated. Default settings are preconfigured for the Account, Contact, and Prospect business components to support integration to ODQ Matching Server (IIR), but you can configure the mappings to your requirements or to support integration to other vendors.

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

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

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

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

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

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

2 In the Vendor List, select the record for the required vendor.

3 Click the BC Vendor Field Mapping view tab.

4 In the BC Operation list, select the record for the required business component and Data Cleansing operation.

The field mappings are displayed in the Field Mapping list.

5 In the Field Mapping list enter the required values for Business Component Field and Mapped Field.

Configuring Business Components for Data Matching Using the Matching Server

This topic gives one example of configuring a business component for data matching with SSA. You might use this feature differently, depending on your business model. For example, see the following sample procedures:

- "Using Siebel Business Applications to Configure a Business Component for Data Matching with SSA" on page 60
- "Using Siebel Tools to Configure a Business Component for Data Matching with SSA" on page 61

Using Siebel Business Applications to Configure a Business Component for Data Matching with SSA

Use the following procedure to configure a business component for data matching with SSA using the Administration - Data Quality screen in your Siebel application.

To configure a business component for data matching with SSA using the Administration - Data Quality screen in the Siebel application

1. Navigate to the Administration - Data Quality screen, then the Third Party Administration view.
2. In the Vendor List, select the record with the name SSA.
3. Click the Vendor Parameter view tab.
4. In the Vendor Parameter list, create new records with the parameter names and values provided in the following table.

Name	Value	Comments
Business_component_name DeDup Record Type	*Business_component_name*	
SSA Match Purpose	Enter one of the following values: ■ Company_Mandatory ■ Company_Optional ■ Contact_Mandatory ■ Contact_Optional **NOTE:** 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 "Match Purpose" on page 67.

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

For more information, see "Mapping Data Matching Vendor Fields to Siebel Business Components" on page 58.

Using Siebel Tools to Configure a Business Component for Data Matching with SSA

Use the following procedure to configure a business component for data matching with SSA using Siebel Tools.

To configure a business component for data matching with SSA using Siebel Tools

1 Start Siebel Tools.

2 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 (DBA) to assist you with the design and creation of the key table. The table can be created and applied from Siebel Tools for test purposes. Because the table can become very large, it is recommended that your DBA move it to a specific disk or tablespace.

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.

b Create a new business component using 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*.

3 Create a link and the Algorithm Type field for the business component and the key business component you created in Step 2 on page 61.

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 2 on page 61 with the properties and values provided in the following table.

Property	Value
Name	DeDuplication - SSA *Business Component name* Key
Destination Link	*Business Component name*/DeDuplication - SSA *Business Component name* Key
Destination Business Component	DeDuplication - SSA *Business Component name* Key

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

- Name is set to DeDuplication - SSA Account Key
- Destination Link is set to Account/DeDuplication - SSA Account Key
- Destination Business Component is set to DeDuplication - SSA Account Key

c Navigate to the Field object type and create a new field for the business component you created in Step 2 on page 61 with the properties and values provided in the following table.

Property	Value
Name	Algorithm Type
Multivalue Link	DeDuplication - SSA *Business Component name* 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*.

4 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 2 on page 61.

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.

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

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

	Value		
Property	**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.

5 Create a DeDuplication Results business component using the S_DEDUP_RESULT table with the field values shown in the following table.

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 CRM 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 CRM 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.

6 Add the new DeDuplication Results 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.

7 Configure an applet as your DeDuplication Results List Applet using the business component you configured in Step 2 on page 61. 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 might 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.

8 To trigger real-time matching on a new applet, 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 List Applet and specify the applet that you configured in Step 7 on page 64 in the value column.

9 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.

10 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 92.

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

Configuring the Windows Displayed in Real-Time Data Matching

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

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

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

- "Changing a Window Name" on page 65
- "Adding a Deduplication Window for an Applet" on page 65

Where child applets are concerned, see "Configuring Real-Time Deduplication Window for Child Applets" on page 66.

Changing a Window Name

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

To change the name of the window displayed

1. 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.
5. Recompile the SRF.
6. Restart the Siebel Server.

Adding a Deduplication Window for an Applet

Use the following procedure to add a deduplication window for an applet.

To add a deduplication window for an applet

1. 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. Recompile the SRF.
6. Restart the Siebel Server.

Configuring Real-Time Deduplication Window for Child Applets

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

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

To configure the real-time deduplication window for a child applet (Account Contact view)

1. In Siebel Tools, query for the following applet:

   Account Contact List Applet

2. Add the following user property to this applet:
   - Name: DeDuplication Results Applet
   - Value: DeDuplication Results (Contact) List Applet
3. Recompile the SRF.
4. 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 15 on page 52 shows the preconfigured mandatory fields that Oracle Corporation 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 must repeat Step 2 of this procedure.

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

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.

Each of the Account, Contact, and List Mgmt Contact Prospect business components has an associated SSA *Business_component_name* Match Purpose vendor parameter 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 17.

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 17. Match Purpose Values for SSA Match Purpose Vendor Parameters

Match Purpose	SSA Field	Mandatory or Optional
Company_Mandatory	Company	Mandatory
	Address	Mandatory
	Address2	Mandatory
	Zip	Optional
	ID	Optional
Company_Optional	Company	Mandatory
	Address	Optional
	Address2	Optional
	Zip	Optional
	ID	Optional

Table 17. Match Purpose Values for SSA Match Purpose Vendor Parameters

Match Purpose	SSA Field	Mandatory or 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. Navigate to the Administration - Data Quality screen, then the Third Party Administration view.
2. In the Vendor List, select the record for the SSA vendor.
3. Click the Vendor Parameters view tab.
4. In the Vendor Parameters List, select the required SSA Match Purpose vendor parameter, and set the value as required.

Siebel Data Quality User Properties

This topic provides information about deduplication and data cleansing user properties. It includes the following topics:

- Deduplication User Properties on page 69
- Data Cleansing User Properties on page 71

Deduplication User Properties

Deduplication detects possible matches to records in specified business components during record creation and update. The matching process begins with the Dedup token, which is an identifier calculated for each account, contact, or prospect in the database as well as the newly created or modified record. Using the value of the Dedup token, the Siebel application passes to the data quality matching engine a short list of prequalified possible matches for further refinement.

Deduplication, like data cleansing, is configured in two business component user properties, but also affects certain views and applets. The deduplication feature is disabled or enabled for the application through settings in the .cfg file. After being turned on at the application level, deduplication can be turned off for a specific business component by deactivating all of the child user properties (by setting the Inactive property to TRUE). Deduplication cannot be turned off for individual records. If you configure a business component for deduplication, it must also be configured for data cleansing and data cleansing must be turned on. (The reverse is not necessarily true; you can configure data cleansing for a business component without configuring deduplication.)

Data deduplication works only on applets that use the CSSFrameBase and CSSFrameListBase classes, and classes derived from these. Data cleansing works for applets that use any class.

NOTE: Components from the vendor’s Corporation must be installed for this functionality to work.

The following deduplication user properties are described in this topic:

- “DeDup Token Value” on page 69
- “DeDuplication CFG File” on page 70
- “DeDuplication Field n” on page 70
- “Deduplication Results BusComp” on page 71
- “Deduplication Results List Applet” on page 71

DeDup Token Value

This user property allows you to specify the dedup token calculation expression for a business component.

DeDuplication CFG File

This user property stores the name of the vendor configuration file used for account deduplication.

Parent Object Type	Business Component
Description	Vendor CFG file used for Account deduplication.
Functional Area	Data Cleansing

DeDuplication Field n

This user property sets up a correspondence between a vendor Connector data field and a Siebel CRM business component data field.

Parent Object Type

Business Component

Description

Sets up a correspondence between a vendor Connector data field and a Siebel CRM business component data field. The value consists of a pair of quoted strings in double quotation marks, separated by a comma, with the first string identifying the vendor field name and the second string identifying the Siebel CRM name.

The set of fields mapped in DeDuplication Field user properties is the set of fields that is passed in records in the candidate set to the vendor Connector. The candidate set consists of records with a dedup token exactly or partially matching the calculated dedup token of the record being added or modified, and therefore representing possible duplicates.

Functional Area

Data Quality

Deduplication has a set of numbered user properties that set up correspondences between vendor fields and Fields in Business Components. These field mapping properties have names of the form DeDuplication Field *n*, where *n* is an integer value (for example, DeDuplication Field 7). The syntax for the Value property in a DeDuplication Field user property is the same as for a DataCleansing Field user property: the value consists of a pair of quoted strings in double quotation marks, separated by a comma, with the first string identifying the vendor field name and the second string identifying the Siebel CRM name. The set of fields mapped in DeDuplication Field user property is the set of fields that is passed in records in the candidate set to the vendor Connector. The candidate set consists of records with a dedup token exactly or partially matching the calculated dedup token of the record being added or modified, and therefore representing possible duplicates.

The Prospects business component requires additional user property configuration beyond that required for Contacts and Accounts. Prospects share name processing capabilities in the vendor Connector with Contacts, and Contact data (rather than Prospect data) is assumed by the system to be present. In order to specify that Prospect data is being processed, two additional user properties must be added, DeDuplication Results business component and DeDuplication Results applet.

For information about setting numbered instances of a user property, see *Siebel Developer's Reference*.

Deduplication Results BusComp

This user property stores the name of the business component that will hold returned deduplication data.

Parent Object Type	Business Component
Description	This user property name is added to the Account, Contact, or Prospects business component upon which deduplication is being performed. The value in the user property name is the name of the business component that will hold the returned data, for example, DeDuplication Results (Prospect).
Functional Area	Data Quality

Deduplication Results List Applet

This user property stores the name of the pick applet that prompts the user to resolve duplicates.

Parent Object Type	Applet
Value	The name of the pick applet used to prompt the user to resolve duplicates, for example: ■ DeDuplication Results (Account) List Applet ■ DeDuplication Results (Contact) List Applet
Usage	Add the DeDuplication Results List Applet user property to the applet from which you want to trigger real-time deduplication.
Functional Area	Data Quality

Data Cleansing User Properties

The following data cleansing user properties are described in this topic:

- "DataCleansing Field n" on page 71
- "DataCleansing Type" on page 72

DataCleansing Field *n*

This user property allows you to specify a correspondence between a field name in the vendor Connector and a field name in the Siebel application.

Parent Object Type	Business Component

Value

The value for the DataCleansing Field user property uses the following syntax:

```
“[Vendor Field Name]”, “[Siebel Field Name]”
```

For example, the following value specifies a correspondence between the Firm Location field in the Vendor Connector and the Primary Account Location field in the Siebel application.

```
"Firm Location", "Primary Account Location"
```

Usage

Used for Siebel Data Quality, which performs address verification, name and address standardization, and duplicate record identification, in real-time and batch modes.

NOTE: All data quality user properties require components from the vendor’s Corporation.

See also *Siebel Developer’s Reference*.

DataCleansing Type

This user property allows you to specify to the vendor Connector what kind of data is being validated in the Data Cleansing Field.

Parent Object Type

Business Component

Value

The value for the DataCleansing Type user property must be one of the following:

- Contact   indicates that the data consists of person name records.
- Account   indicates that data consists of business or office name records.
- Address   indicates that data consists of postal addresses.

All types have capitalization validated.

Usage

Data cleansing operates differently on each of these types. For example, business components with Address cleansing have reconciliation performed between address fields and the ZIP (Postal Code) field.

7 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:

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 called whenever a user attempts to save a new or modified account, contact, or prospective contact record to the database.

For data cleansing, the fields configured for data cleansing are standardized before the record is committed.

For data matching, when 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 srvrmgr command-line utility to submit server component batch jobs. Depending on business requirements and the amount of new and changed records, you can run these batch jobs at intervals.

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 78.

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 80.

Real-Time Data Matching and Cleansing

In real-time mode, SDQ is called when you save a new or modified record. If both data cleansing and data matching are enabled for the same object manager, data cleansing runs first.

If data cleansing is enabled, a set of fields preconfigured to use data cleansing are standardized before the record is committed.

If data matching is enabled, and the new record is a potential duplicate, one of the following dialog boxes appears:

- Duplicate Accounts
- Duplicate Contacts
- Duplicate Prospects

You must then decide the fate of the new record, as follows:

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

  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 commits the new record to the database. The record that you choose becomes the surviving record, and is saved, but is then deleted by the merge process. Merging is performed as described in "Sequenced Merges" on page 91.

In real-time mode, if you enter two new records that have the same Name and Location, then an error message displays similar to the following: *The same values for (Name, Location) already exist. To enter a new record, make sure that field values are unique.* Real-time data matching prevents creation of a duplicate record in the following ways:

- If you are in the process of creating a new record, that record is not saved.
- If you are in the process of modifying a record, the change is not made to the record.

NOTE: Only certain fields are configured to support data matching and data cleansing. If you do not enter values in these fields when you create a new record, or you do not modify the values in these fields when changing a record, data cleansing and data matching are not triggered. For more information about which fields are preconfigured for different business components, see "Preconfigured Field Mappings for SSA" on page 149.

Batch Data Matching and Data Cleansing

Batch processing provides a means to cleanse and match a large number of records at one time. You can run batch jobs as stand-alone tasks or schedule batch tasks to run on a recurring basis.

After the Data Quality Manager server component (DQMgr) is enabled and you have restarted the Siebel Server, you can start your data quality tasks.

You can start and monitor tasks for the Data Quality Manager server component in one of the following ways:

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

You can specify a data quality rule in the batch job parameters. This is a convenient way of consolidating and reusing batch job parameters and also of overriding vendor parameters. For more information, see "Data Quality Rules" on page 87.

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 CRM scripting on business components used for data matching (such as Account, Contact, List Mgmt Prospective Contact, and so on), the modifications to the fields by the script execute in the background and might not trigger logic that activates user interface features. For example, the scripting might not trigger UI features such as pop-up windows that show potential matching records.

Data Quality Batch Job Parameters

Table 18 shows the parameters used in Data Quality batch jobs. The names of the parameters for both Data Quality Manager component jobs and srvrmgr commands are given.

Table 18. Data Quality Batch Job Parameters

Job Parameter or Server Manager Parameter	Required	Description
Buscomp name bcname	Yes	The name of the business component: Possible values include: ■ Account ■ Contact ■ List Mgmt Prospective Contact ■ Business Address - applicable to Data Cleansing operations only. For Siebel Industry Applications, CUT Address is used instead of Business Address.
Business Object Name bobjname	Yes	The name of the business object. Possible values include: ■ Account ■ Contact ■ List Mgmt ■ Business Address - applicable to Data Cleansing operations only
Operation Type opType	Yes	The type of operation: Possible values are: ■ Data Cleansing - cleanses data ■ Key Generate - generates match keys ■ Key Refresh - refreshes match keys ■ DeDuplication - performs data matching.
Object Sorting Clause objsortclause	No	Applicable to Data Matching operations only. Indicates how candidate records are sorted for optimal processing by the data matching software. The default value is Dedup Token.

Table 18. Data Quality Batch Job Parameters

Job Parameter or 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: ■ **First value.** 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. ■ **Second value.** Applicable to the Universal Connector only. It specifies whether the job is a full or incremental data matching job. ■ **Third value.** This is obsolete. Enter an empty string. For more information about the use of DQSetting, see "Matching Data Using Batch Jobs" on page 80.
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 Table 14.
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 Table 14 on page 45.
Threshold	No	Specifies a value for the Threshold data quality parameter. This is applicable to SSA only. For more information about this parameter, see Table 14.

Table 18. Data Quality Batch Job Parameters

Job Parameter or Server Manager Parameter	Required	Description
Rule Name	No	Specifies the name of a data quality rule. A rule with the specified name must have been created in the Administration - Data Quality screen, Rules view. For example: `RuleName="Rule_Batch_Account_Dedup"` For more information, see "Data Quality Rules" on page 87.

Cleansing Data Using Batch Jobs

The following procedure describes how to use a batch job to perform data cleansing on records in a selected business component.

To effectively exclude selected records when running data cleansing tasks, you must add the following command to your object WHERE clause:

```
[Disable DataCleansing] <> 'Y'
```

CAUTION: When you run a process in batch mode, any visibility limitation against your targeted data set is ignored. It is recommended that you allow only a small group of people to access the Siebel Server Manager to run your data quality tasks, otherwise you run the risk of corrupting your data.

To perform batch mode data cleansing

1. Start the Server Manager Program.
2. At the srvrmgr prompt, enter a command like one of those in the following table to perform data cleansing. The following 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="'','',''"
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'"

Business Component	Example of Server Manager Command
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	run task for comp DQMgr with bcname=Account, bobjname=Account, opType="Key Generate", objwhereclause="[Updated] > '07/18/2005 16:00:00'"
	Refresh	run task for comp DQMgr with bcname=Account, bobjname=Account, opType="Key Refresh", objwhereclause="[Name] LIKE '*search_string**'"
Contact	Generate	run task for comp DQMgr with bcname=Contact, bobjname=Contact, opType="Key Generate", objwhereclause="[Updated] > '07/01/2005 14:10:00'"
	Refresh	run task for comp DQMgr with bcname=Contact, bobjname=Contact, opType="Key Refresh", objwhereclause="[Last Name] LIKE '*search_string**'"
List Mgmt Prospective Contact	Generate	run task for comp DQMgr with bcname="List Mgmt Prospective Contact", bobjname="List Mgmt", opType="Key Generate", objwhereclause="[Updated] > '07/18/2005 16:00:00'"
	Refresh	run task for comp DQMgr with bcname="List Mgmt Prospective Contact", bobjname="List Mgmt", opType="Key Refresh", objwhereclause="[Last Name] LIKE '*search_string**'"

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

- The generation commands generate keys for all records in the business component that have been updated since the specified date and time.
- The refresh commands refresh keys for all records in the business component that match the search string in the specified field.

You can use either of these two types of WHERE clauses for both generation and refresh operations.

If you want to generate or refresh keys for all records in the business component, use a WHERE clause containing a wildcard character (*) to match all records, as follows:

```
objwhereclause="[field_name] LIKE '*'"
```

Matching Data Using Batch Jobs

Depending on your business requirements, you might want to use batch jobs to perform data matching on some or all of the records in the supported business components. If you must run a data matching batch job on all the records in a business component, the work can often be completed more quickly by splitting the work into a number of smaller batch jobs (not more than 50,000 to 75,000 records at a time). When data matching has been performed on all of the records in the business component, you can run future data matching batch jobs on just the new or changed records.

If you want to perform data matching for some number of mutually-exclusive subsets of the records in a business component, such as all the records where a field name starts with a given letter, use a separate job to specify each subset, with WHERE clauses as follows:

```
objwhereclause="[field_name] LIKE 'A*'"
objwhereclause="[field_name] LIKE 'B*'"
...
objwhereclause="[field_name] LIKE 'Z*'"
objwhereclause="[field_name] LIKE 'a*'"
...
objwhereclause="[field_name] LIKE 'z*'"
```

The following example further describes batch data matching.

Example of Batch Data Matching Using the Universal Connector

You must run batch mode key generation on all existing records before you run real-time data matching. The Universal Connector requires generated keys in the key tables first before you can run real-time data matching. The key generation is done within the deduplication task, which is the reason for running deduplication on all existing records first. For more information about batch data cleansing and matching, see "Batch Data Matching and Data Cleansing" on page 75.

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 79.
2. At the srvrmgr prompt, enter commands like those in the following table to perform data matching.

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

Full Data Matching Jobs

In a full data matching job, the records for which you want to locate duplicates and the candidate records that can include those duplicates are defined by the same search specification. A full data matching job is specified with the value Yes in the DQSetting parameter, see Table 19 on page 82.

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, bobjname=Account, opType=DeDuplication, objwhereclause="[Name] LIKE
'A*'"
```

Jobs like this that perform data matching for a subset of records are still considered to be full data matching jobs because the data to be checked does not depend on earlier data matching.

Incremental Data Matching Jobs

If you want to perform data matching for some number of nonexclusive subsets of the records in a business component, such as all the records that have been created or updated since you last ran data matching, use a WHERE clause that includes an appropriate timestamp, and also adjust the DqSetting clause of the command as shown in Table 19.

Table 19. DqSetting Parameter Details and Sample Values

DqSetting Parameter Sequence	Valid Values	Comments
First section	Leave blank	Specify as two adjacent quotation marks.
Second section (Enforce Search Spec on Candidate Records)	■ Yes ■ No (default)	Specifies whether or not the same search specification is used for both the records whose duplicates are of interest and the candidate records that can include those duplicates. ■ Use Yes for full data matching batch jobs. ■ Use No for incremental data matching batch jobs.
Third section	Leave blank	None.

This kind of job is considered an *incremental data matching* job, because data matching was done earlier and does not 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',''",
bcname=Account, bobjname=Account, opType=DeDuplication, objwhereclause="[Updated]
> '08/18/2005 20:00:00'
```

NOTE: If you do not specify the DQSetting parameter, or leave the second value of the DQSetting parameter blank, the job will be an incremental data matching job.

Customizing Data Quality Server Component Jobs for Batch Mode

Rather than specifying parameters each time you start a data quality batch job, you can customize the Data Quality Manager server component with the parameters that you require. This is mainly for ease of use when starting tasks using the srvrmgr program.

You use the Administration - Server Configuration views to create customized components (depending on the Data Quality Manager Server component). You specify Data Quality Manager as the Component Type. Sample customization settings are shown in Table 20 on page 84 through Table 23 on page 86. 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: For Siebel CRM Version 7.8 or later, you can also set specific parameters for a data quality task and save the configuration as a template by using the Administration - Server Configuration screen, Job Templates view. The benefit in doing so is that there is no need to copy component definitions. For more information about Siebel application 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 20 on page 84 through Table 23 on page 86 to allow easier location of log files.

Sample Component Customization for Data Matching

Table 20 through Table 22 provide the recommended custom component definitions for Account, Contact, and Prospect objects for the SDQ Matching Server.

Table 20. 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 21. 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 21. 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 22. Recommended Custom Component Definitions for SDQ Matching Server for Prospects

Component Alias	Component Name	Description	Component Parameter	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

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

Sample Component Customization for Data Cleansing

Table 23 provides the recommended custom component definitions for the Account, Contact, Prospects, and Business Address business objects.

Table 23. Recommended Custom Component Definitions for Data Cleansing

Component Alias	Component Name	Description	Component Parameter	Value
DQMgrAcctDClns	DQ Account Data Cleansing	Data quality data cleansing for accounts	Buscomp Name	Account
			Business Object Name	Account
			Operation Type	Data Cleansing
DQMgrContDClns	DQ Contact Data Cleansing	Data quality data cleansing for contacts	Buscomp Name	Contact
			Business Object Name	Contact
			Operation Type	Data Cleansing
DQMgrPrspDClns	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
DQMgrAddrDClns	DQ Address Data Cleansing	Data quality data cleansing for addresses	Buscomp Name	Business Address
			Business Object Name	Business Address
			Operation Type	Data Cleansing

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

Data Quality Rules

In the Administration - Data Quality screen, Rules view, you can define rules for each of the data quality operations that are performed in real-time and in batch mode.

The data quality rules specify the parameters used when a data quality operation is performed in real-time or in batch mode. For example, you can create a rule for the batch mode Data Cleansing operation on the Account business component for a particular vendor. The parameters used are the vendor parameters defined for the applicable vendor, but you can override these parameters by specifying the equivalent rule parameters. However, the values set for Key Type, Match Threshold, and Search Type in the User Preferences data quality settings override the equivalent rule parameters.

You can only create rules for business components for which data cleansing or data matching are supported. This includes the preconfigured business components and any additional business components that you configure for data cleansing and data matching. Also, you can only create rules for operations that are supported for a particular vendor. For example, you cannot define data cleansing rules for SSA. For each vendor, the supported operations and business components are defined in the Administration - Data Quality screen, Third Party Administration view.

You can create only one real time rule for each combination of vendor, business component, and operation name. However, you can create any number of batch rules for each combination of vendor business component, and operation name.

When you define a rule for real time mode, the rule is applied each time data cleansing or data matching is performed for the business component. When you define a rule for batch mode, the rule is applied if you specify the name of the rule in the batch job parameters, see "Data Quality Batch Job Parameters" on page 76. Using rules in this way allows you to consolidate batch job parameters into a reusable rule.

You can specify a search specification, business object name, business component name, threshold, and operation Type in a rule or in the job parameters when you submit a job in batch mode. The values in the job parameters override any value in the rules.

NOTE: Do not confuse data quality rules with the matching rules that are used by the third party software.

Creating a Data Quality Rule

Use the following procedure to create a data quality rule.

To create a data quality rule

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

2 Create a new record. Some of the fields are as follows.

Field	Comments
Name	Enter a unique name for the rule.

Field	Comments
Search Specification	Enter a search specification. Applicable for Operation Type Batch only.
Vendor Name	Select a vendor name, for example, SSA.
Operation Type	Select Batch or Real Time.
Operation Name	Select one of the following: ■ Data Cleansing ■ DeDuplication ■ Key Generate (batch mode only) ■ Key Refresh (batch mode only)
Threshold	Enter a value between 50 and 100. This value overrides the value in the Data Quality settings. Applicable for Operation Name DeDuplication only.
Source Business Component	Select a business component name.
Source Business Object	Select the business object name corresponding to the business components

An example of a rule is shown in the following table. This is a rule for DeDuplication operations for all Account records whose name starts with Aa.

Field	Value
Name	Rule_Batch_Account_Dedup
Search Specification	[Name] LIKE 'Aa*'
Vendor Name	SSA
Operation Type	Batch
Operation Name	DeDuplication
Threshold	60
Source Business Component	Account
Source Business Object	Account

3 (Optional) Specify rule parameters.

   a Click the Rule Parameter view tab.

   b Create rule parameters by selecting a parameter and entering the required value.

Merge Algorithm in the Object Manager Layer

The Merge Records functionality is used by customers to enhance data quality. For example, duplicate accounts might be merged to a target account or you might want to merge duplicate opportunities. To call the feature, select two or more records and choose Edit, and then Merge Records from the application-level menu. For more information about the Merge Records menu option, see *Siebel Fundamentals*.

Example of the Merge Records Process

You want to merge accounts A1 and A2 into A1. These accounts might have child quotation marks or associated contacts. These relationships are defined using one-to-many or many-to-many links. The following describes what happens after the merge:

- Account A2 is deleted after the merge.
- The contacts associated with A2 are associated with A1 after the merge.

The links defined between the business components are used to implement the merge algorithm. The algorithm used by the merge process at the OM layer is explained below for one-to-many and many-to-many links.

Overview of Merge Algorithm

The following topic provides a brief overview of what happens during the merge process (and uses the example of merging accounts A1 and A2 into A1).

The merge process starts by enumerating through all link definitions that might be relevant, for example, in the case of the example, where the source business component is accounts.

One-to-Many Relationship

A one-to-many relationship defines the destination field, which is the foreign key in the detail table that points to a row in the parent table. Only links where the source field is "Id", that is, where the foreign key in the detail table stores the ROW_ID of the parent table row, are considered.

To make children of A2 point to A1, the merge must update the destination field in the detail table to now point to the ROW_ID of A1.

> **User property name:** Use Literals for Merge
> **Use Literals For Merge**: S_BU
> **Value**: TRUE

When merging two records, the child records of the loser record point to the survivor record and the LAST_UPD and LAST_UPD_By columns of those child records are also updated. For example, account A2 is merged to account A1. Account A2 has service request SR1, and SR2. The columns LAST_UPD, and LAST_UPD_BY of SR1 and SR2 are updated during merge process.

From the example, link account/quote foreign key in S_DOC_Quote is account Id (TARGET_OU_ID). TARGET_OU_ID stored the ROW_ID of the A2. It is now updated to point to ROW_ID of A1.

SQL generated:

```
UPDATE S_DOC_QUOTE set TARGET_OU_ID = 'Row Id of A1'
```

where:

TARGET_OU_ID is equal to 'Row Id of A2'

While the merge is processing the link account or quote, it also checks to see if there are other foreign keys from quote pointing to account using the join definitions. These keys are also updated.

An optimization is used to ensure that there are no redundant update statements. For example, if there are two links defined (account or quote and account or quote with primary with the same destination field Account Id), the process would update TARGET_OU_ID of S_DOC_QUOTE twice to point to A1. To avoid this scenario, a map of table name or column name of the processed field is maintained. The update is skipped if the column has been processed before.

After the update you might have duplicate children for an account. For example, if the unique key for a quote is the name of the quote, merging two accounts with quotation marks of the same name will result in duplicates. The CONFLICT_ID column of children that will become duplicates after the merge is updated. This operation is performed before the actual update.

The user must examine duplicate children (identified by CONFLICT_ID being set) to make sure that they are true duplicates. For example, if the merged account has child quotation marks named Q1 and Q1, it is possible that these refer to distinct quotation marks. If this is the case, the name of one of the quotation marks must be updated and the children must be merged.

Many-to-Many Relationship

The many-to-many relationship (Accounts-Contacts) differs slightly from the one-to-many relationship in that it is implemented using an intersection table that stores the ROW_IDs of parent-child records. On a merge, the associations must be updated. The Contacts associated with the old Account must now be associated with the new Account.

The Inter parent column of the intersection table is updated to point to the new parent. As in the one-to-many case, to avoid redundant updates, a map of intersection tables that have been processed is maintained. Therefore, if the source and target business components have the same base table, both child and parent columns are updated.

The CONFLICT_ID column of intersection table entries that become duplicates after the merge is updated.

In contrast to the one-to-many link case, duplicates in the intersection table imply that the same child is being associated with the parent two or more times. However, there might be cases where the intersection table has entries besides the ROW_ID of the parent and child rows that store information specific to the association.

The duplicate association records are only preserved when records are determined as unique, according to the intersection table unique key. This means those duplicate association records might have some unique attributes and these attributes are part of a unique key of the intersection table. CONFLICT_ID does not account for uniqueness among records.

Merging 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.

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

You can merge duplicate records in the following ways:

- **Merge Records option (Edit, Merge Records)**. Performs the standard merge functionality available in Siebel Business 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 *Siebel 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 91. This action also performs a cleanup in the appropriate Deduplication Results table to remove the unnecessary duplicate records. This is the preferred method for deduplicating account, contact, and prospect records.

Sequenced Merges

You use a sequenced merge to merge multiple records into one record. You assign sequence numbers to the records so that the record with the lowest sequence number becomes the surviving record, and the other records, the nonsurviving records, are merged with the surviving record.

When records are merged using a sequence merge, the following rules apply:

- All non-NULL fields from the surviving record are kept.

  Any fields that were NULL in the surviving record are populated by information (if any) from the nonsurviving records. Missing fields in the surviving record are populated in ascending sequence number order from corresponding fields in the nonsurviving records.
- The children and grandchildren (for example, activities, orders, assets, service requests, and so on) of the nonsurviving records are merged by associating them 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 number field is empty, and a contact record with a Sequence number of 3 has a value of Work Phone number. If the field Email address and Work Phone number 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 number is taken from the record of sequence number 3.

A sequence number is required for each record even if there are only two records.

Field Characteristics for Sequenced Merges

A field must have the following specific characteristics to be eligible for use in a sequenced merge:

- The field 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 Record Duplicates" on page 92

   This involves creating a query to find a subset of the duplicate records and then review the query results. For example, you might want to create a query that includes a subset of all duplicate records where the Name field starts with the letter A.

2. "Merging Duplicate Records" on page 93

   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 Matching and Data Cleansing" on page 75.

Filtering Record Duplicates

Use the following procedure to filter duplicate records. This task is a step in "Process of Merging Duplicate Records" on page 92.

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 94.

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. This task is a step in "Process of Merging Duplicate Records" on page 92.

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.

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 91.

Merging Child Duplicate Records

Use the following procedure to merge child duplicate records.

To merge child duplicate records

1. In the appropriate Duplicate *XXX* Resolution view, enter 1 in the Sequence field for the parent record.

2. Enter 2 and so on in the Sequence field for each of the child duplicate records.

3. Select the records to be merged, and select the parent records last.

4. Click Merge.

Using Fuzzy Query

To run a query using fuzzy query, this facility must be enabled and several conditions must be met as described in "Enabling and Disabling Fuzzy Query" on page 51.

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 52.
- The query must leave optional fields blank.

If the conditions for fuzzy query are not satisfied, then any queries you make use standard query functionality.

Using Fuzzy Query for Accounts

Use the following procedure to use fuzzy query for accounts.

To use fuzzy query for accounts

1. Navigate to the Accounts screen, then the Accounts List view.
2. Click the Query button.
3. Enter your query, and then click Go.

   The query results contain fuzzy matches in addition to regular query matches.

Using Fuzzy Query for Contacts

Use the following procedure to use fuzzy query for contacts.

To use fuzzy query for contacts

1. Navigate to the Contacts screen, then the Contacts List view.
2. Click the Query button.
3. Enter your query, and then click Go.

   The query results contain fuzzy matches in addition to regular query matches.

Using Fuzzy Query for Prospects

Use the following procedure to use fuzzy query for prospects.

To use fuzzy query for prospects

1. Navigate to the List Management screen, then the Prospects view.
2. Click the Query button.
3. Enter your query, and then click Go.

   The query results contain fuzzy matches in addition to regular query matches.

Example of Enabling and Using Fuzzy Query with Accounts

This topic gives an example of enabling and using fuzzy query. You might use this feature differently, depending on your business model.

In the following example, you enable fuzzy query for accounts, and then enter the query criteria. The query results contain fuzzy matches from the DeDuplication business service in addition to regular query matches.

NOTE: EAI Siebel Adapter does not support fuzzy queries. In addition, scripting does not support fuzzy queries.

To enable and use fuzzy query for accounts

1. Perform the steps in "Enabling and Disabling Fuzzy Query" on page 51
2. Perform the steps in "Enabling Siebel Data Quality at the User Level" on page 49.

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

3. Navigate to the Accounts screen, then the Account list view.
4. Enter your query, and then click Go.

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

   Up to 10 records having Name set to Symphony are displayed.

   NOTE: If the number of Symphony account records is fewer than 10, then the fuzzy query results includes records where symphony is lowercase (as well as uppercase). For example, if four records for *Symphony* and 100 records for *symphony* are found in the database, the fuzzy query result shows four *Symphony* records and six *symphony* records. However, if fuzzy query is disabled, only the four *Symphony* records appear.

Calling Data Matching and Data Cleansing from Scripts or Workflows

This topic provides information about calling data matching and data cleansing methods from external callers such as scripts or workflows.

You can call data quality from external callers to perform data matching. You can use the Value Match method of the Deduplication business service to:

- Match data in field or value pairs against the data within Siebel business components
- Prevent duplicate data from getting into the Siebel application through non-UI data streams

You can also call data quality from external callers to perform data cleansing. There are preconfigured Data Cleansing business service methods—Get Siebel Fields and Parse. Using an external caller, such as scripting or a workflow process, you first call the Get Siebel Fields method, and then call the Parse method to cleanse contacts and accounts.

The following scenarios provide more information about calling data quality from external callers:

- "Scenario for Data Matching Using the Value Match Method" on page 96
- "Scenario for Data Cleansing Using Data Cleansing Business Service Methods" on page 97

Scenario for Data Matching Using the Value Match Method

This topic gives one example of how you can call the Value Match business service method using Siebel Workflow. You can use the Value Match method differently, depending on your business model.

In this scenario, a company 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 calls the Value Match method.

In this case, the implementation calls the Value Match method as a step in the workflow process that adds the contact. This step matches incoming contact information against the contacts within the Siebel database. To prevent the introduction of duplicate information into the Siebel application, the implementation adds processing logic to the script 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 call and use the Value Match method, see "Value Match Method" on page 97.

Scenario for Data Cleansing Using Data Cleansing Business Service Methods

This topic gives one example of how you can call the Data Cleansing business service methods using Siebel Workflow. You might use the methods differently, depending on your business mode.

A system administrator or data steward in an enterprise wants to cleanse data before it enters the data through EAI or EIM interfaces. To do this, the system administrator or data steward uses a script or workflow that cleanses the data. The script or workflow calls the Get Siebel Fields method, which returns a list of cleansed fields for the applicable business component. Then the script or workflow calls the Parse method, which returns the data for the cleansed fields.

For information about how to call and use the Get Siebel Fields and Parse methods, see "Data Cleansing Business Service Methods" on page 101.

Deduplication Business Service Methods

This topic describes the following Deduplication business service method: "Value Match Method" on page 97.

NOTE: For information about other deduplication business service methods that are available, see *Siebel Tools Online Help*.

"Scenario for Data Matching Using the Value Match Method" on page 96 gives one example of how you can call the Deduplication business service Value Match method.

Value Match Method

You can use the Value Match method of the Deduplication business service to find potential matching records in the Siebel application or when you want to prevent duplicate data from getting into the Siebel application through non-UI data streams.

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

Arguments

The Value Match method consist of input and output arguments, some of which are property sets. Table 24 describes the input arguments, and Table 25 on page 99 describes the output arguments.

CAUTION: The Value Match method arguments are specialized. Do not configure these components.

Table 24. Value Match Method Input Arguments

Name	Type	Property Name	Description	Comments
Adapter Settings **NOTE:** Adapter Settings and Match Values are child property sets of the input property set.	Property Set	Code Page	Code page	Optional. Applicable only to SSA. Allowed values for the Adapter Settings property set are SSA-specific, but you can use the same allowed values as in SSA third-party configuration. The value Override can be specified to override the corresponding setting information obtained by the service from the administration screens, vendor properties, and so on.
		Population	Population values.	
		Search Level	The search level	
		Threshold	The threshold score for a duplicate record. A match is considered only if the score exceeds this value.	

Match Values
NOTE: Adapter Settings and Match Values are child property sets of the input property set.
	Property Set	*Business component field names, and value* pairs:
		`<Name1><Value1>,`
		`<Name2><Value2>,`
		`<Name3><Value3>,`
		`...`
			The matched business component's field name and the corresponding field value:
			`(Last Name, 'Smith')`
			`(First Name, 'John'),`
			`and so on ...`
			NOTE: Each pair must be a child property set of Match Values.
				These name-value pairs are used as the matched value rather than the current row ID of the matched business component. The vendor field mappings for the matched business component are used to map the business component field names to vendor field names.

BC Name	Property	BC Name	The name of the matched business component.	Required.
Update Modification Date	Property	Update Modification Date	If set to N, the match modification date is not updated.	Optional. The default is Y.
Use Result Table	Property	Use Result Table	If set to N, matches are not added to the result table. Instead, matches are determined by the business service.	Optional. The default is Y.

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 25.

CAUTION: The Value Match method arguments are specialized. Do not configure these components.

Table 25. Value Match Method Output Arguments

Name	Type	Property Name	Description	Comments
End Time	Property	End Time	The run end time.	
Match Info **NOTE:** Match Info is a child property set of the output property set.	Property Set	Matchee Row ID	The row ID of a matching record.	If you match against existing records, the record ROW_IDs are found and returned in the Match Info property set.
		Score	The score of a matching record.	
Num Added Results	Property	Num Added Results		
Num Candidates	Property	Num Candidates	The total number of potential matches if scores are not used.	
Num Results	Property	Num Results	The number of actual matches.	
Row Value	Property	Row Value	The row ID of the match or matches found.	
Start Time	Property	Start Time	The run start time.	

Called From

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

Example

The following is an example of using Siebel eScript to call the Value Match method. This script calls the Value Match method to look for duplicates of John Smith from the Contact business component and then returns matches, if any. After the script finishes, determine what you want to do with the duplicate records, that is, either merge or remove them.

```
function Script_Open ()

{
    TheApplication().TraceOff();
    TheApplication().TraceOn("sdq.log", "Allocation", "All");
    TheApplication().Trace("Start of Trace");

    // Create the Input property set and a placeholder for the Output property set
    var svcs;
    var sInput, sOutput, sAdapter, sMatchValues;
    var buscomp;

    svcs = TheApplication().GetService("DeDuplication");
    sInput = TheApplication().NewPropertySet();
    sOutput = TheApplication().NewPropertySet();
    sAdapter = TheApplication().NewPropertySet();
    sMatchValues = TheApplication().NewPropertySet();

    // Set Generic Settings input property parameters
    sInput.SetProperty("BC Name", "Contact");
    sInput.SetProperty("Use Result Table", "N");
    sInput.SetType("Generic Settings");

    // Set Match Values child input property parameters
    sMatchValues.SetProperty("Last Name", "Smith");
    sMatchValues.SetProperty("First Name", "John");
    sMatchValues.SetType("Match Values");
    sInput.AddChild(sMatchValues);

    // Set Adapter Settings child input property parameters
    sAdapter.SetProperty("Search Level", "Narrow");
    sAdapter.SetProperty("Population", "Default");
    sAdapter.SetType("Adapter Settings");
    sInput.AddChild(sAdapter);

    // Invoke the "Value Match" business service
    TheApplication().Trace("Property set created, ready to call Match method");
    svcs.InvokeMethod("Value Match", sInput, sOutput);

    // Get the Output property set and its values
    TheApplication().Trace("Value Match method invoked");
    var propName = "";
    var propVal = "";
    propName = sOutput.GetFirstProperty();
    while (propName != "")
    {
    propVal = sOutput.GetProperty(propName);
    TheApplication().Trace(propName);
    TheApplication().Trace(propVal);
    propName = sOutput.GetNextProperty()

    }
    TheApplication().Trace("End Of Trace");
    TheApplication().TraceOff();
}
```

Data Cleansing Business Service Methods

This topic describes the following data cleansing business service methods:

- "Get Siebel Fields Method" on page 101
- "Parse Method" on page 102

"Scenario for Data Cleansing Using Data Cleansing Business Service Methods" on page 97 gives one example of how you can call the data cleansing business service methods.

Get Siebel Fields Method

Get Siebel Fields is one of the methods of the Data Cleansing business service. This method returns a list of cleansed fields for a given business component.

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

Arguments

Get Siebel Fields arguments are listed in Table 26.

Table 26. Get Siebel Fields Arguments

Argument Name	Display Name	Input/ Output	Data Type	Required?	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 the corresponding values are Field Name.

Usage

This method is used with the Parse method in the process of cleansing data in real time, and it is used with the Parse All function in the process of using a batch job to cleanse data.

Called From

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

Parse Method

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

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

Arguments

Parse arguments are listed in Table 27.

Table 27. Parse Arguments

Argument Name	Display Name	Input/ Output	Data Type	Required?	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.

Called From

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

For more information about Siebel Workflow, see *Siebel Business Process Designer Administration Guide*.

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 39 and "Specifying Data Quality Settings" on page 43.

- **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 49.

- **Third-party software.** Verify that the third-party software is installed and you have followed all instructions from the third-party installation documents.

If you have configured new business components for data cleansing or data matching, also check the following:

- **Business component Class property.** Verify that the business component Class property is CSSBCBase.

- **Vendor Properties.** Verify that the vendor parameters and vendor field mappings have the correct values and that the values are formatted correctly. For example, there must be a space after a comma in vendor properties that have a compound value. *Siebel System Administration Guide*

TIP: Check My Oracle Support regularly for updates to troubleshooting and other important information. For more information about My Oracle Support, see "Information about SDQ on My Oracle Support" on page 195.

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

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 CRM 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 28. Split the tasks into smaller tasks and run them concurrently.

Table 28. 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 78.

Optimizing Data Matching Performance

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

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

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

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

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

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

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

For more information about running batch data matching, see "Matching Data Using Batch Jobs" on page 80.

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" on page 107
- "Data Quality Manager Server Tasks" on page 108
- "Data Quality Settings" on page 109

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 29 and work with a database administrator to make sure there is sufficient space available for these tables.

  Table 29. Table Size Consideration

Table	Sizing Consideration
S_PER_DEDUP_KEY S_ORG_DEDUP_KEY S_PRSP_DEDUPKEY	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: ■ **Limited key type.** Between two and four times more records ■ **Standard key type.** Up to an estimated six times more records
S_DEDUP_RESULT	After a full deduplication run, this table can contain five to six times the number of records in the three base tables combined. ■ If a Typical or Exhaustive search type is used, more records are inserted into the results table. ■ If a low match threshold is used, the matching process generates a larger number of records that are inserted into the results table.

- For the DB2 DBMS, 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 DBMS, 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:

  ```
  ObjWhereClause="[Last Name] < 'A'"
  ObjWhereClause="[Last Name] like 'A*'"
  ObjWhereClause="[Last Name] like 'a*'"
  ObjWhereClause="[Last Name] like 'B*'"
  ObjWhereClause="[Last Name] like 'b*'"
  ...

  ObjWhereClause="[Last Name] like 'Y*'"
  ObjWhereClause="[Last Name] like 'y*'"
  ObjWhereClause="[Last Name] like 'Z*'"
  ObjWhereClause="[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 *Siebel 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)`

To Query for...	Key Generation Example	Data Matching Example

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 using 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 set to 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 79.

For more information about running batch data matching jobs, see "Customizing Data Quality Server Component Jobs for Batch Mode" on page 83.

Data Quality Settings

The following are recommended values for optimal performance for the settings in the Administration - Data Quality screen, 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 43.

A Setting Up Oracle Data Quality Matching Server for Data Matching

This appendix is for customers who intend to use Oracle Data Quality Matching Server for data matching. Oracle Data Quality Matching Server uses a licensed version of the third-party software, Informatica Identity Resolution (IIR), for data matching.

The integration uses Universal Connector in a mode where match candidate acquisition takes place within ODQ Matching Server. Since the match keys are generated and stored within ODQ Matching Server, key generation and key refresh operations are eliminated within Siebel CRM.

This integration, whereby match candidate acquisition takes place within ODQ Matching Server, cannot be used by other third-party data quality matching engines.

The following topic is included in this appendix:

- Process of Setting Up ODQ Matching Server for Data Matching on page 111

NOTE: For more information about SSA-NAME3 software, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.

Process of Setting Up ODQ Matching Server for Data Matching

To set up the ODQ Matching Server for data matching, perform the following tasks:

1. "Setting Up the Environment and the Database" on page 112

   This includes the subtask "Creating Database Users and Tables for ODQ Matching Server" on page 113

2. "Installing ODQ Matching Server" on page 116
3. "Configuring ODQ Matching Server" on page 121
4. "Configuring the Siebel Application for ODQ Matching Server" on page 124
5. "Obtaining the ODQ Matching Server License Key" on page 126
6. "Applying the Universal Data Quality Patch" on page 127
7. "Modifying Configuration Parameters for ODQ Matching Server" on page 128
8. "Deploying and Activating Workflows for ODQ Matching Server Integration" on page 129
9. "Initial Loading of Siebel Data into ODQ Matching Server Tables" on page 130
10. "Synchronizing Siebel Data with ODQ Matching Server Tables" on page 131

Setting Up the Environment and the Database

This task is a step in "Process of Setting Up ODQ Matching Server for Data Matching" on page 111. This topic describes the prerequisites that are needed before starting to set up ODQ Matching Server for data matching, and also how to set up an Oracle database for ODQ Matching Server.

Java Runtime Environment

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

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

For example, on a Win32 client:

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

On UNIX operating system:

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

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

Network Protocol

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

ODBC Driver

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

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

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

A configuration block has the following syntax:

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

Table 30 on page 115 shows some example odbc.ini configurations.

NOTE: ODQ Matching Server provides a custom driver for the Oracle database that is installed during the installation of the product. ODQ Matching Server does not use the standard driver shipped with the Oracle DBMS.

Creating Database Users and Tables for ODQ Matching Server

Creating database users and tables involves executing a number of scripts on the ODQ Matching Server database. The scripts that you must execute are located in the Oracle Data Quality Applications media pack on Oracle E-Delivery. Once the media pack is downloaded and installed, the scripts are located in the NM3_2807_XXX folder where *XXX* is the name of the operating system.

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

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

  For more information about testing the connectivity on UNIX, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.

To create database users and tables for ODQ Matching Server if using an Oracle database

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

   NOTE: You must be logged in as database administrator to execute 1_x.sql.

2. Log in to the database as the new database user (created in Step 1 with appropriate privileges to create and update IIR tables), then execute the following SQL scripts to create other IIR database tables, such as IDT and IDX tables. You can execute the following SQL scripts in any order:

   NOTE: IDT tables store the copy of source records in the IIR database. IDX tables store the index keys for IDT tables. Each IDT table can have one or more IDX tables associated with it.

   a. Execute 2_idstbora.sql to create control tables for ODQ Matching Server.

   b. Execute 3_updsyncu.sql to create database objects required by IIR to synchronize data in ID tables with updates to user source tables.

      Run this script on all databases containing user source tables that require synchronization, and also before loading any ID tables that require synchronization.

   c. Execute 4_updsynci.sql to create database objects required by ODQ Matching Server to synchronize data in ID tables with updates to user source tables.

      Run this script on the database which will contain IDTs, and also before loading any ID tables that require synchronization.

   d. Execute 5_updsyncg.sql to create database objects required by ODQ Matching Server to synchronize data in SSA-ID tables with updates to user source tables.

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

      NOTE: You must be logged in as database administrator to execute _updsyncg.sql.

Table 30 on page 115 shows some example odbc.ini configurations.

Table 30. Example odbc.ini Configuration Blocks

Database	Description	Example odbc.ini Configurations

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

Table 30. Example odbc.ini Configuration Blocks

Database	Description	Example odbc.ini Configurations

Microsoft SQL Server	Microsoft provides a Windows ODBC driver named sqlsrv32. It is configured by adding a new Data Source Name (DSN) by navigating to Control Panel, Administrative Tools, Data Sources (ODBC).
	For more information about the sqlsrv32 driver, see the appropriate Microsoft manuals for specific details.
	The ODBC_Driver name is sqlsrv32 and the Native_DB_Service is the server name (-S parameter of the osql and bcp utilities).
	The SQL Server Native Client (sqlncli.dll) can be used as an alternative driver.
		`[production]`
		`DataSourceName = msq2003`
		`ssadriver = sqlsrv32`
		`server = mySQLServer`

Universal Database (UDB)	For more information about the db2cli and db2 drivers, see the appropriate UDB manuals for full details.
	UDB must be installed prior to the installation of the ODQ Matching Server.
		`[test-udb]`
		`DataSourceName = udb8`
		`ssadriver = db2cli`
		`ssaunixdriver = db2`
		`server = UDB_database_alias`

Sybase	For more information about the sybdrvodb drivers, see the appropriate Sybase manuals for installation specifics.	`[production]`
		`DataSourceName = ase150`
		`ssadriver = sybdrvodb`
		`ssaunixdriver = sybdrvodb`
		`server = mySybaseServer`

Installing ODQ Matching Server

This task is a step in "Process of Setting Up ODQ Matching Server for Data Matching" on page 111. The following tasks describe the steps involved in installing ODQ Matching Server on Microsoft Windows and UNIX operating systems respectively.

- "Installing ODQ Matching Server on Microsoft Windows" on page 117
- "Installing ODQ Matching Server on UNIX" on page 117

NOTE: Before installing and setting up ODQ Matching Server, install SSA-NAME3 server. For more information about SSA-NAME3 server installation on Microsoft Windows and on UNIX, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.

Installing ODQ Matching Server on Microsoft Windows

Use the following procedure to install ODQ Matching Server on Microsoft Windows.

To install ODQ Matching Server on Microsoft Windows

1. Run setup from the root directory of the installed product media pack (which you downloaded from Oracle E-Delivery), and follow the onscreen prompts.
2. Test ODBC connectivity.

   If this is your first time installing ODQ Matching Server, then an ODBC configuration file named `%SSABIN%\odbc.ini` is created. To generate this file, the installer will prompt you for the necessary information. If this is not your first time installing ODQ Matching Server and you have a configuration file from a previous installation, then the odbc.ini file will be copied to the server's bin directory.
3. Run the IIR Console in Configure Mode.

   This mode tests the installation by loading a predefined IIR system and verifying its search and synchronization results.
4. Apply bug fixes.

   Confirm that all bug fixes are installed by using the version command in the command prompt from the IIR server installation's bin directory ($SSABIN). For example:

   ```
   c:\ids\iss2704s\bin>version
   Identity Systems' ISS v2.7.04 (FixCD083 FixK057) + FixK066 + FixK104
   ```
5. Start the IIR server, for example, by navigating to:

   Programs, Identity Systems' Products, ISS 2.7.04, ISS v2.7.04 Server Start

For more information about each of these steps and about IIR server installation on Microsoft Windows, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.

Installing ODQ Matching Server on UNIX

The installation process for ODQ Matching Server on UNIX is different from the installation process on Microsoft Windows (for example, there is no installation wizard). Installation on UNIX mainly consists of unarchiving the release, setting up the environment variables, and then running the console.

Use the following procedure to install ODQ Matching Server on UNIX. The installation can be completed from any user account as root privilege is not required.

To install ODQ Matching Server on UNIX

1 Unarchive the release.

The ODQ Matching Server UNIX release is supplied as a compressed tar file. Copy the compressed tar file to your UNIX computer. If using FTP, binary mode must be selected. Once the compressed tar file is on your UNIX computer:

a If the packaged release file has a .Z file name extension, then extract the file as follows: `compress -d xxx.tar.z`.

If the packaged release file has a .zip file name extension, then extract the file as follows: `unzip xxx.zip`.

b Extract the binaries from the tar archive. For example:

```
tar -xf xxx.tar
```

After this has completed, there will be a directory on your computer called iss2704.ful containing all of the IIR components.

c Install the latest fix pack (if there is one available). A fix pack is distributed as a compressed tar file. Decompress and untar fixes, using one of the following:

```
compress -d fixknnnsrv.tar.z
tar -x fixKnnnsrv.tar

unzip fixKnnnsrvt.zip
tar -x fixKnnnsrv.tar
```

d Optionally, rename the release directory. For example:

```
mv iss2704.ful iss2704
```

NOTE: If applying a fix pack in the future, you must temporarily restore the release directory name because untarring the fix pack places updates in the iss2704.ful directory.

2 Set up the environment variables.

a Start a fresh Bourne shell, for example: sh.

b Edit the `$SSATOP/ssaset` script where SSATOP represents the name of the IIR release directory, as follows:

NOTE: The most recent version of ssaset is available on each fix pack and is called ssaset.ori. Copy this file to ssaset and then customize as required. Most of the environment variables that are set by this script do not need modification. However, some must be set and others must be checked.

- ❑ Set SSATOP using an absolute path, and point it to the IIR Installation directory:

```
/home/user/iss/iss2704
```

- ❑ Set SSAN3V2TOP to the SSA-NAME3 v2 directory.

```
/home/user/iss/nm32704
```

❏ Establish your environment to enable communication with your host DBMS using the scripts provided to you by your database administrator. These scripts typically set some environment variables and paths required to communicate with your DBMS. The setup script for an Oracle Database is usually called by:

```
. oraenv
```

❏ Select the relevant setup script for your operating system and source it. This sets environment variables used to link the DBMS library. For example:

```
. $SSATOP/setups/xxx
```

where xxx refers to your operating system. For example:

```
. $SSATOP/setups/gcclibc2 (Linuz)
. $SSATOP/setups/rs32lfio (AIX 4.3.3, 32 bit)
. $SSATOP/setups/solaris (Oracle Solaris 8, 32 bit)
. $SSATOP/setups/hpux11 (HP-UX 11.0 PA-RISC, 64 bit)
```

❏ Create the `$HOME/tmp` directory (if not already created) for each user who runs IIR programs and scripts.

The setup script sets the variable SSATEMP to point to the `$HOME/tmp` by default. This directory is used for storing various temporary files. Alternatively, you can change this environment variable to point to another directory as long as a unique directory name is used for each user.

❏ Check which version of awk is available on your UNIX operating system, then change the SSAAWK statement to specify this version. Use nawk for Oracle Solaris.

❏ Set SSAPR to reflect the location of the SSA-NAME3 Population Rule directory (that is, if it differs from the default setting).

❏ Set your Java environment if you want to use any of the Java clients, including the Console, from UNIX. The CLASSPATH and PATH variables must be set up appropriately (for more information, see "Setting Up the Environment and the Database" on page 112). Make sure Java Help (jhall.jar) is included in the CLASSPATH. For example:

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

❏ Modify the line which sets SSAORATOP to point to the Oracle home directory ($ORACLE_HOME) on your system.

❏ Select the appropriate LD_LIBRARY_PATH (or the equivalent) for your operating system. Comment out the inappropriate paths for different operating systems.

❏ The default library paths work for most users. However, it is recommended that users change Library Paths to suit their requirements (for more information, see "Configuring ODQ Matching Server on UNIX" on page 122).

❏ Modify SSASQLLDR to specify the name of your Mass Load utility.

For an Oracle Database, it is called sqlldr.
For UDB, it is called db2.
For Microsoft SQL, it is called bcp.
On OS/390, you must use `$SSABIN/ssaload`.

Make sure that dbtype is set or changed accordingly. For example, use the following for Oracle Databases:

```
SSASQLLDR="sqlldr"
SSA_DB_TYPE="ora"
```

- Check the host and port information for each IIR server program. Port numbers only must be changed if the default ports clash with those used by an existing process. The default port numbers and corresponding environment variables are listed in the following table.

Server Name	Environment Variable
Search Server	SSA_SEPORT=1666
Connection Server	SSA_COPORT=1667
Rulebase Server	SSA_RBPORT=1668
Connection Server	SSA_CSPORT=1669

- The OS/390 environment requires additional variables.

  SSA_DB_PLAN and SSA_DB_SUBSYSTEM must be set to the name of your database plan and subsystem respectively.

  DB2HLQ must be set to the high level qualifier of your DB2 installation dataset (usually DSN710).

- SSA_DB_RBSTORAGE must be set to the storage clause of the Rulebase (for example: "IN SSA02").

c Source ssaset to establish the environment variables. For example:

```
.$SSATOP/ssaset
```

Note the following:

- POSIX compliant shells can terminate the script when any undefined environment variables are referenced. If this occurs, add the following command as the second line of the script and then source it again:

```
set +u
```

- Change the following line:

```
SSA_XSHOST="$SSA_XSHOST:$SSA_XSPORT"
```

  to:

```
SSA_XSHOST="$SSA_XXHOST:$SSA_XSPORT"
```

- Before the following line:

```
echo "ssaset: The ISS environment variables have been set.":
```

  Add the following environment parameters:

```
SSA_RBNAME="odb:0:<ISSDB_userName>/<ISSDB_password>@<Service_Name>";
export SSA_RBNAME

SSA_RB_RESTART_ID="0"; export SSA_RB_RESTART_ID
```

where:

- ISSDB_userName is the database user created in SQL script 1_x.sql.
  ISSDB_passWord is the database password created in SQL script 1_x.sql.
  Service_Name is the connection string name as defined in the odbc.ini file.

3 Test the installation by running a regression test to confirm that the software has been installed and configured correctly.

  For more information about testing the installation on UNIX, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.

4 Start and stop the IIR server from a UNIX shell prompt as follows:

```
$SSABIN/idsup
$SSABIN/idsdown
```

5 Verify that IIR is operational. For example, from within the IIR ConsoleClient:

  a Click Search Client, select the Search Client radio button, then click OK.

  b Click the appropriate search mechanism button, input the desired search criteria, then click Search.

    The records stored in the IIR tables are returned.

For more information about each of these steps and about IIR server installation and set up on UNIX, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.

Configuring ODQ Matching Server

This task is a step in "Process of Setting Up ODQ Matching Server for Data Matching" on page 111. The following tasks describe the steps involved in configuring ODQ Matching Server on Microsoft Windows and UNIX operating systems respectively.

- "Configuring ODQ Matching Server on Microsoft Windows" on page 121
- "Configuring ODQ Matching Server on UNIX" on page 122

Configuring ODQ Matching Server on Microsoft Windows

Use the following procedure to configure ODQ Matching Server on Microsoft Windows.

To configure ODQ Matching Server for data matching on Microsoft Windows

1 Modify the odbc.ini file located at <drive>:`\IIR Installation folder\iss2704s\bin\` to contain the ODBC connection string of your target database, for example, as follows:

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

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

Table 30 on page 115 summarizes the ODBC drivers required for different operating systems.

2 Copy the SiebelDQ.sdf file to the following (IIR server) folder location:

```
<Drive>:\IIR Installation folder\iss2704s\ids
```

3 Turn on XML Sync Server by modifying the idsenvs.bat file located in `<Drive>:\IIR Installation Folder\iss2704s\bin`.

In idsenvs.bat, activate the following commands (by removing the "::" at the beginning of the line):

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

4 Create a tmp folder for the IIR Synchronizer Workflow Log in `<Drive>:\IIR Installation Folder\iss2704s\`. For example:

```
C:\ids\iss2704s\tmp
```

NOTE: If you install IIR on a different drive (other than `C:\`), you must modify the ISSErrorHandler workflow in the Siebel application to specify the correct log folder. Other modifications that must be made if you install IIR on a drive other than `C:\` include modifying action sets and the location where you deploy the XML files.

5 Start IIR Server, for example, by navigating to:

Programs, Identity Systems' Products, ISS 2.7.04, ISS v2.7.04 Server Start (Configure Mode)

6 Start the IIR Console Client (in Admin Mode), for example, by navigating to the following:

Programs, Identity Systems' Products, ISS 2.7.04, ISS v2.7.04 Console Client (Admin Mode)

7 Create new system in IIR using SiebelDQ.sdf.

The system that you create in IIR (Console Client, Admin Mode) must hold all the IDT and IDX database tables. For more information about creating a new system in IIR, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.

8 When the system is created (initially, it will be empty), run LoadIDT from the IIR Console Client. For more information, see "Initial Loading of Siebel Data into ODQ Matching Server Tables" on page 130.

Configuring ODQ Matching Server on UNIX

Use the following procedure to configure ODQ Matching Server on UNIX.

To configure ODQ Matching Server for data matching on UNIX

1 Copy the most recent version of the shared library libssaiok.so (libssaiok.sl on HP-UX) to the SSA-NAME3 bin directory.

If the version packaged with IIR is more recent than the one packaged with SSA-NAME3, copy the ssaiok shared library from the IIR server distribution to the SSA-NAME3 bin directory as follows:

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

No action is required if the version packaged with IIR is older than the one packaged with SSA-NAME3.

2 Set the shared library path according to your operating system.

The following table shows examples of shared library paths.

Operating System	Shared Library Path
Linux and Oracle Solaris	`LD_LIBRARY_PATH="$SSABIN:$SSANM3BIN:`*`<DBMS shared object location>`*`" export LD_LIBRARY_PATH`

HP-UX	`SHLIB_PATH="$SSABIN:$SSANM3BIN:`*`<DBMS shared object location>`*`"`
	`export SHLIB_PATH`

AIX	`LIBPATH="$SSABIN:$SSANM3BIN:`*`<DBMS shared object location>`*`"`
	`export LIBPATH`

3 Modify the odbc.ini file to contain the ODBC connection string of your target database:

a Copy the odbc.ini.ori file located in the $SSATOP/bin folder, and rename it odbc.ini.

b Edit the odbc.ini to contain the ODBC connection string of your target database, for example, as follows:

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

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

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

Table 30 on page 115 describes the ODBC drivers required for different operating systems.

NOTE: The odbc.ini file is located in the $SSATOP/bin folder.

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

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

```
unzip - sysdeffile.zip
```

For more information about configuring ODBC on UNIX, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery

Configuring the Siebel Application for ODQ Matching Server

Configuring the Siebel application for ODQ Matching Server involves carrying out tasks in Siebel Tools and your Siebel application. This topic is a step in "Process of Setting Up ODQ Matching Server for Data Matching" on page 111.

To configure the Siebel application for ODQ Matching Server, complete the steps in the following procedures:

1. "Configuring Siebel Tools for ODQ Matching Server" on page 124
2. "Configuring the Siebel Application for ODQ Matching Server" on page 125

Configuring Siebel Tools for ODQ Matching Server

Configuring Siebel Tools for ODQ Matching Server involves creating a new project for IIR in Siebel Tools, importing sif files (business service, integration objects, and workflow sif files) to it, and modifying some business components. To configure Siebel Tools for ODQ Matching Server, complete the steps in the following procedure.

To configure Siebel Tools for ODQ Matching Server

1. Create a new project for IIR, then import the business service, integration objects, and workflow sif files with overwrite mode.

   a. Create a new IIR project in Siebel Tools as follows:

      ❑ In the Object Explorer, select the Project object type.

      ❑ In the Object List Editor, right-click and choose New Record.

      ❑ Enter a Name for the project (in this case, IIR) and then step off the record.

   b. Import the IIR sif files into Siebel Tools as follows:

      ❑ Click Tools - Import from Archives.

      ❑ Click Yes on the dialog that opens.

      ❑ On the Import Wizard - Preview window that opens, select the sif file, the option to Overwrite the object definition in the repository, then Next to import the sif file.

      Repeat this step, as required. Import sif files in the following sequence:

      BusinessService.sif
      IntegrationObjects.sif
      Workflows.sif

      For more information about creating new projects in Siebel Tools and importing objects from an archive file into Siebel Tools, see *Siebel Tools Online Help*.

   c. After importing the business service and integration objects, compile them into the srf file.

d. After importing the workflows, deploy and activate them.

   For more information about deploying and activating workflows, see "Deploying and Activating Workflows for ODQ Matching Server Integration" on page 129, and also *Siebel Business Process Framework: Workflow Guide*.

2. Modify the Contact business component and List Mgmt Prospective Contact business component.

   a. Select the business component you want to change.

   b. Click Tools, then Lock Project.

   c. Then create new fields or modify existing fields to the business component, as required:

   Change the Contact business component by modifying the field shown in the following table.

Name	Calculated	Calculated Value	Force Active
First Name Last Name	Yes	[First Name] + " " + [Last Name]	Yes

   Change the List Mgmt Prospective Contact business component by adding the new fields shown in the following table.

Name	Calculated	Calculated Value	Force Active
Prospect Id	Yes	[Id]	-
First Name Last Name	Yes	[First Name] + " " + [Last Name]	Yes

   d. When finished modifying business components, compile them into the srf file.

Configuring the Siebel Application for ODQ Matching Server

Configuring your Siebel application for ODQ Matching Server involves registering IIR as an SDQ connector and creating administration data (such as, vendor parameters and field mappings) in Siebel for ODQ Matching Server. To configure your Siebel application for ODQ Matching Server, complete the steps in the following procedure.

To configure your Siebel application for ODQ Matching Server

1. Copy the ssadq_cfg.xml file to the `siebsrvr\SDQConnector` folder. This file contains the global configuration parameters for IIR.

2. Copy the operating system-specific IIR libraries to the `siebserv\bin` folder (Siebel dll, sscaddsv.dll, are already included in the patch). For example:

   On Windows, copy the following IIR connector DLLS to `siebserv\bin`:

   ```
   ssadq.dll
   ssaiok.dll
   ssasec.dll
   ```

   On Linux (and other UNIX-like systems), copy the following IIR connector libraries to siebserv/lib:

```
libssadq.so
libssaiok.so
libssasec.so
```

3. Register IIR as a new SDQ connector.

   For example, navigate to Administration - Data Quality screen, then the Third Party Administration view, and create a new record with the information shown in the following table.

Vendor Name	DLL Name
ISS	ssadq

   For more information about connector registration, see "Registering New SDQ Connectors" on page 54.

4. Set up the following business component-to-operation field mappings for IIR, as shown in the following table.

Business Component	Operation
Account	DeDuplication
Contact	DeDuplication
List Mgmt Prospective Contact	DeDuplication

   For more information about setting up field mappings for vendors in the Siebel application, see "Mapping of Vendor Fields to Business Component Fields" on page 57.

5. Verify that the preconfigured ODQ Matching Server vendor parameter and field mapping values as described in "Universal Connector Parameter and Field Mapping Values for ODQ Matching Server" on page 136 are set up.

6. Verify that your Siebel application integration objects, action sets, and run-time events are set up so that they are in sync with IIR tables. For more information about synchronization, see "Synchronizing Siebel Data with ODQ Matching Server Tables" on page 131.

Obtaining the ODQ Matching Server License Key

This task is a step in "Process of Setting Up ODQ Matching Server for Data Matching" on page 111.

The *Oracle Data Quality Matching Server License File*, or universal license key, is a special license file that is located in the Oracle Data Quality Applications media pack on Oracle E-Delivery. You must apply the Oracle Data Quality Matching Server License File before using SSA-NAME3 and IIR. This can be done either before or after installing the product.

For instructions about how to install the license file for the ODQ Matching Server, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.

When installing ODQ Matching Server on Windows, note that you must:

- Use the -a switch to extract the license key file.
- Copy the license key file, ssalcns.key, to the following location:

  `<drive>\ids\nm32704\pr`

When installing ODQ Matching Server on UNIX, note that you must:

- Compress the license key file and then use FTP to copy it to the UNIX server.
- Use the -a switch to extract the license key file.
- Copy the license key file, ssalcns.key, to the following location:

  `%SSAN3V2TOP%\pr`

NOTE: For more information about the Oracle Data Quality Matching Server License File, create a service request (SR) on My Oracle Support. Alternatively, you can phone Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers are listed on My Oracle Support.

Applying the Universal Data Quality Patch

You must apply the Universal Data Quality (UDQ) patch in order to obtain the new Siebel DLL (sscaddsv.dll) and the appropriate IIR connector DLLs (ssadq.dll, ssaiok.dll, and ssasec.dll).

CAUTION: It is recommended that you make a backup copy of your existing sscaddsv.dll file first.

To apply the UDQ patch, complete the steps in the following procedure. This procedure is a step in "Process of Setting Up ODQ Matching Server for Data Matching" on page 111.

To apply the UDQ patch

1. Download the UDQ patch from My Oracle Support, and install it.

   You must install both Siebel Server and Siebel Tools. The repository patch (\reppatch) is only available when you install Siebel Tools.

   After installing the UDQ patch, the new Siebel DLL is located in the `<Siebel Server>\bin` folder, and all other files (for example, readme, sif, and IIR connector DLLs) are located in the `<Siebel Tools>\reppatch\DQ_ISS_Integration` folder.

2. Refer to IIR configuration files located in the `<Siebel Tools>reppatch\DQ_ISS_Integration\ISS_Configuration` folder, and configure IIR as described in "Configuring ODQ Matching Server" on page 121.

3. Refer to Siebel configuration files located in the `<Siebel Tools>\reppatch\DQ_ISS_Integration\Siebel_Configuration` folder, and configure the Siebel application for IIR as described in "Configuring the Siebel Application for ODQ Matching Server" on page 124.

Modifying Configuration Parameters for ODQ Matching Server

This procedure is a step in "Process of Setting Up ODQ Matching Server for Data Matching" on page 111.

The ssadq_cfg.xml file contains the global configuration parameters for ODQ Matching Server (IIR). To modify ssadq_cfg.xml, complete the steps in the following procedure.

To modify configuration parameters for the ODQ Matching Server

1. Open up a text editor.
2. Modify the following parameters in the ssadq_cfg.xml file, as required:
   a. Set `iss_host` to point to the server where IIR is running.
   b. Set `iss_port` to 1667 (which is the default), unless you are using a different port for installation.
   c. Set the `rulebase_name` parameter. For example, with Oracle Database 10*g*:
      - username is ssa
      - password is SIEBEL
      - ServiceName is Target (As specified in the odbc.ini file for the IIR server)
      - `rulebase_name` is odb:0:ssa/SIEBEL@Target

      For more information about the format of the rulebase name, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.
   d. Set `contact_system`, `account_system`, and `prospect_system` to the name of the system created the ODQ Matching Server (IIR) using the SiebelDQ.sdf file.

      The system that you create in IIR (Console Client, Admin Mode) must hold all the IDT and IDX database tables. For more information about creating a new system in IIR, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.

      NOTE: If you want to run IIR against only a single entity (for example, Accounts) as opposed to multiple entities (Accounts, Contacts, and Prospects), then you must alter the definitions within the SiebelDQ.sdf file to include only the one entity that you want as otherwise the synchronizer fails to run. In this example, you must remove the definitions for Contacts and Prospects.
3. Save the ssadq_cfg.xml file and copy to the SDQConnector folder on Siebel Server for changes to take effect:

```
siebsrvr\SDQConnector
```

Deploying and Activating Workflows for ODQ Matching Server Integration

This task is a step in "Process of Setting Up ODQ Matching Server for Data Matching" on page 111.

In the Siebel application, you must deploy and activate the following workflow processes for real-time integration of ODQ Matching Server:

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

These workflows are used in building data files for the following:

- "Initial Loading of Siebel Data into ODQ Matching Server Tables" on page 130
- "Synchronizing Siebel Data with ODQ Matching Server Tables" on page 131

For more information about deploying and activating workflows, see *Siebel Business Process Framework: Workflow Guide*.

Initial Loading of Siebel Data into ODQ Matching Server Tables

This procedure is a step in "Process of Setting Up ODQ Matching Server for Data Matching" on page 111. To initially load your Siebel application data into ODQ Matching Server (IIR) tables, complete the steps in the following procedure.

To initially load Siebel application data into ODQ Matching Server tables

1. In the Siebel application:

   a. Log in as administrator and navigate to Administration - Runtime Events screen, then the Action Sets view.

   b. Query in the Name field for ISSLoad* and make sure that all action sets are active. If some are not:

      - ❑ Activate them by selecting the Active check box for each action set.
      - ❑ Then reload run-time events by clicking Menu, and selecting Reload Runtime Events.

   c. Navigate to Administration - Data Quality screen, then the Data Quality Settings view, and modify any one of the records (for example, increase the Match Threshold value) to trigger the export data process.

      This action triggers the run-time events to export account, contact, and prospect records from the Siebel application into xml data files. Depending on the number of records in your database, this process can take some time so wait until the process completes.

      NOTE: The location of the XML data file is specified by the ISS Set File Name value of each ISSLoad * action set.

   d. Copy the exported files to the following IIR folder location:

      ```
      <Drive>:\ids\iss2704s\ids\data
      ```

      This is where the Load IDT process gets the files and loads them into IIR.

      NOTE: If the Siebel application and IIR are installed on the same box, then there is no need to copy the exported files to the IIR folder as the export process places the files directly in the IIR folder.

   e. Navigate back to Administration - Runtime Events, then the Action Sets view, and:

      - ❑ Deactivate all ISSLoad* action sets.
      - ❑ Then reload run-time events by clicking Menu, and selecting Reload Runtime Events.

   f. Navigate back to Administration - Data Quality, then the Data Quality Settings view, and reset Match Threshold to it original value.

      This ensures that this action will not be triggered every time a data quality setting is modified.

2 In Informatica Identity Resolution:

   a Start the IIR server by navigating to:

   Programs, Identity Systems' Products, ISS 2.7.04, ISS v2.7.04 Server Start (Configure Mode)

   b Start the IIR Console Client (in Admin Mode) by navigating to:

   Programs, Identity Systems' Products, ISS 2.7.04, ISS v2.7.04 Console Client (Admin Mode)

   c If not already done so, create a new system in IIR using SiebelDQ.sdf. Or, if a system already exists, select it and refresh it by clicking the System/Refresh button.

   The system that you create in IIR (Console Client, Admin Mode) must hold all the IDT and IDX database tables. For more information about creating a new system in IIR, see the relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.

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

   d Load IIR with the data files exported from the Siebel application by clicking the System/Load IDT button.

Synchronizing Siebel Data with ODQ Matching Server Tables

This procedure is a step in "Process of Setting Up ODQ Matching Server for Data Matching" on page 111.

Siebel application account and contact data must be kept in sync with data that is stored in ODQ Matching Server (IIR) tables. This ensures accuracy when populating the match results.

Siebel application integration objects (Account, Contact, and List Mgmt Prospective Contact) are used to send data from the Siebel application to IIR. IIR in turn writes, edits, or deletes records from IIR tables when a record is created, modified, or deleted from the Account, Contact, or Prospect business components.

Siebel application integration objects (Account, Contact, and List Mgmt Prospective Contact) are also used by Siebel Workflows. Workflows are created for loading and sending data according to certain events, and are called when WriteRecord and DeleteRecord events are fired. For this to happen:

- WriteRecord and DeleteRecord events must be created in your Siebel application using the Administration Runtime Events screen, then the Events view.
- Actions (which are attached to the WriteRecord and DeleteRecord events) must be set up in your Siebel application using the Administration Runtime Events screen, then the Action Sets view.

To synchronize your Siebel application data with ODQ Matching Server (IIR) tables, complete the steps in the following procedure.

To synchronize your Siebel application data with ODQ Matching Server tables

1 Set up integration objects (Account, Contact, and List Mgmt Prospective Contact) in your Siebel application.

  This involves importing integration objects, business service, and workflows into Siebel Tools, then compiling them into the srf file as described in "Configuring Siebel Tools for ODQ Matching Server" on page 124.

  a Verify that appropriate action sets are set up for Account, Contact, and List Mgmt Prospective Contact in your Siebel application by navigating to the Administration - Runtime Events screen, then the Action Sets view.

    NOTE: When verifying action set setup, make sure that the IDS_URL profile attribute reflects the URL location of IIR.

    For more information about the action sets that must be set up for Account, Contact, and List Mgmt Prospective Contact, see "Siebel Business Applications Action Sets" on page 171.

    For more information about creating action sets, including creating actions for action sets, see *Siebel Personalization Administration Guide*.

  b Verify that appropriate run-time events are set up in your Siebel application by navigating to Administration - Runtime Events screen, then the Events view.

    The following table describes the run-time events that must be set up for IIR.

    For more information about run-time events, including how to call a workflow process from a run-time event, see *Siebel Business Process Framework: Workflow Guide*.

    For more information about associating events with action sets, see *Siebel Personalization Administration Guide*.

Object Type	Object Name	Event	Action Set Name
BusComp	Data Quality Setting	WriteRecord	ISSLoad Account
BusComp	Data Quality Setting	WriteRecord	ISSLoad Contact
BusComp	Data Quality Setting	WriteRecord	ISSLoad Prospect
BusComp	Account	DeleteRecord	ISSSYNC DeleteRecord Account
BusComp	Contact	DeleteRecord	ISSSYNC DeleteRecord Contact
BusComp	List Mgmt Prospective Contact	DeleteRecord	ISSSYNC DeleteRecord Prospect
BusComp	Account	PreDeleteRecord	ISSSYNC PreDeleteRecord Account
BusComp	Contact	PreDeleteRecord	ISSSYNC PreDeleteRecord Contact
BusComp	List Mgmt Prospective Contact	PreDeleteRecord	ISSSYNC PreDeleteRecord Prospect

Object Type	Object Name	Event	Action Set Name
BusComp	Account	PreWriteRecord	ISSSYNC PreWriteRecord Account
BusComp	Contact	PreWriteRecord	ISSSYNC PreWriteRecord Contact
BusComp	List Mgmt Prospective Contact	PreWriteRecord	ISSSYNC PreWriteRecord Prospect
BusComp	Account	WriteRecord	ISSSYNC WriteRecord Account
BusComp	Contact	WriteRecord	ISSSYNC WriteRecord Contact
BusComp	List Mgmt Prospective Contact	WriteRecord	ISSSYNC WriteRecord Prospect
BusComp	Account	WriteRecordNew	ISSSYNC WriteRecordNew
BusComp	Contact	WriteRecordNew	ISSSYNC WriteRecordNew
BusComp	List Mgmt Prospective Contact	WriteRecordNew	ISSSYNC WriteRecordNew
BusComp	Account	WriteRecordUpdated	ISSSYNC WriteRecordUpdated
BusComp	Contact	WriteRecordUpdated	ISSSYNC WriteRecordUpdated
BusComp	List Mgmt Prospective Contact	WriteRecordUpdated	ISSSYNC WriteRecordUpdated

2 Activate the action sets for Account, Contact, and List Mgmt Prospective Contact, as follows:

   a Navigate to the Administration - Runtime Events screen, then the Action Sets view.

   b Select the Active checkbox for each Action Set that you want to activate.

   c Reload the run-time events by clicking Menu, and selecting Reload Runtime Events.

B Examples of Parameter and Field Mapping Values for Universal Connector

This appendix lists two examples of the preconfigured parameter and field mapping values for the Siebel Data Quality (SDQ) Universal Connector using third-party software. The definitions in this appendix are as preconfigured for ODQ Matching Server (IIR software) and ODQ Address Validation Server.

This appendix includes the following topics:

- About Parameter and Field Mapping Values for Universal Connector on page 135
- Universal Connector Parameter and Field Mapping Values for ODQ Matching Server on page 136
- Universal Connector Parameter and Field Mapping Values for ODQ Address Validation Server on page 139

About Parameter and Field Mapping Values for Universal Connector

Universal Connector definitions are configured as vendor parameters in the Administration - Data Quality screen, Third Party Administration view. Use the following procedure to access and view preconfigured vendor parameters.

CAUTION: Do not reconfigure the parameter settings.

To view the preconfigured vendor parameters

1. Navigate to the Administration - Data Quality screen, then the Third Party Administration view.
2. In the Vendor list, select the record with, for example, the name IIR.
3. Click the Vendor Parameter view tab.

   The vendor parameters are displayed in the Vendor Parameters list.

   You must not reconfigure the parameter settings.

For more information about vendor parameter configuration, see "Configuring Vendor Parameters" on page 57.

The field mappings from vendor fields to Siebel application fields are configured in field mapping parameters in the Administration - Data Quality screen, Third Party Administration view. There are field mappings for each of the supported business components and operations. Use the following procedure to access and view the preconfigured field mappings.

To view preconfigured field mappings

1. Navigate to the Administration - Data Quality screen, then the Third Party Administration view.
2. In the Vendor List, select the record with, for example, the name IIR.

3. Click the BC Vendor Field Mapping view tab.
4. In the BC Operation list, select the record for the required business component and operation.

   The field mappings are displayed in the Field Mapping list.

For information about mapping fields for data matching, see "Mapping of Vendor Fields to Business Component Fields" on page 57.

Universal Connector Parameter and Field Mapping Values for ODQ Matching Server

This topic includes information about the Universal Connector parameter and field mapping values for the ODQ Matching Server (IIR).

- "Preconfigured Vendor Parameters for ODQ Matching Server" on page 136
- "Preconfigured Field Mappings for ODQ Matching Server" on page 137

Preconfigured Vendor Parameters for ODQ Matching Server

Table 31 lists the vendor parameters preconfigured for ODQ Matching Server. You must not reconfigure the parameter settings.

Table 31. Preconfigured ODQ Matching Server Vendor Parameters

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

Preconfigured Field Mappings for ODQ Matching Server

This topic includes information about the preconfigured ODQ Matching Server field mappings for the following business components:

- "Preconfigured Field Mappings for Business Component -Account"
- "Preconfigured Field Mappings for Business Component - Contact"
- "Preconfigured Field Mappings for Business Component - List Mgmt Prospective Contact"

Preconfigured Field Mappings for Business Component -Account

Table 32 shows the ODQ Matching Server data matching field mappings for the Account business component and DeDuplication operation.

Table 32. Preconfigured ODQ Matching Server Field Mappings for Account

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

Preconfigured Field Mappings for Business Component - Contact

Table 33 shows the ODQ Matching Server data matching field mappings for the Contact business component and DeDuplication operation.

Table 33. Preconfigured ODQ Matching Server Field Mappings for Contact

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

Table 33. Preconfigured ODQ Matching Server Field Mappings for Contact

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

Preconfigured Field Mappings for Business Component - List Mgmt Prospective Contact

Table 34 shows the ODQ Matching Server data matching field mappings for the List Mgmt Prospective Contact business component and DeDuplication operation.

Table 34. Preconfigured ODQ Matching Server Field Mappings for List Mgmt Prospective Contact

Business Component Field	Mapped Field
Account	Account
Cellular Phone #	CellularPhone
City	City
Country	Country
Email Address	EmailAddress
First Name Last Name	NAME
Home Phone #	HomePhone
Id	RowId
Middle Name	MiddleName
Postal Code	PostalCode
Social Security Number	SocialSecurityNumber
State	State
Street Address	StreetAddress
Work Phone	WorkPhone

Universal Connector Parameter and Field Mapping Values for ODQ Address Validation Server

This topic includes information about the Universal Connector parameters and field mapping values for the ODQ Address Validation Server.

- "Preconfigured Vendor Parameters for ODQ Address Validation Server" on page 139
- "Preconfigured Field Mappings for ODQ Address Validation Server" on page 139

Preconfigured Vendor Parameters for ODQ Address Validation Server

Table 35 lists the vendor parameters preconfigured for ODQ Address Validation Server.

You must not reconfigure the parameter settings.

Table 35. Preconfigured Vendor Parameters for ODQ Address Validation Server

Name	Value
Account DataCleanse Record Type	Account
Contact DataCleanse Record Type	Contact
List Mgmt Prospective Contact DataCleanse Record Type	Prospect
Personal Address DataCleanse Record Type	Business Address
CUT Address DataCleanse Record Type	Business Address
DQ Send Empty Field To Third Party Vendor	No
DQ Cleanse High Deliverable Address	Yes
Parameter 1	"global", "iss-config-file", "ssadq_cfgasm.xml"

Preconfigured Field Mappings for ODQ Address Validation Server

This topic includes information about the preconfigured ODQ Address Validation Server field mappings for the following business components:

- "Preconfigured Field Mappings for Business Component - Account"
- "Preconfigured Field Mappings for Business Component - Contact"
- "Preconfigured Field Mappings for Business Component - List Mgmt Prospective Contact"

- "Preconfigured Field Mappings for Business Component - CUT Address"
- "Preconfigured Field Mappings for Business Component - Personal Address"

Preconfigured Field Mappings for Business Component - Account

Table 36 shows the data cleansing field mappings for the Account business component and data cleansing operation.

Table 36. Preconfigured Field Mappings for ODQ Address Validation Server Business Component - Account

Business Component Field	Mapped Field
Name	Account.Name

Preconfigured Field Mappings for Business Component - Contact

Table 37 shows the data cleansing field mappings for the Contact business component and data cleansing operation.

Table 37. Preconfigured Field Mappings for ODQ Address Validation Server Business Component - Contact

Business Component Field	Mapped Field
First Name	Contact.First Name
Last Name	Contact.Last Name
Middle Name	Contact.Middle Name

Preconfigured Field Mappings for Business Component - List Mgmt Prospective Contact

Table 38 shows the data cleansing field mappings for the List Mgmt Prospective Contact business component and data cleansing operation.

Table 38. Preconfigured Field Mappings for ODQ Address Validation Server Business Component - List Mgmt Prospective Contact

Business Component Field	Mapped Field
First Name	List Mgmt Prospective Contact.First Name
Job Title	List Mgmt Prospective Contact.Job Title
Last Name	List Mgmt Prospective Contact.Last Name
Middle Name	List Mgmt Prospective Contact.Middle Name

Preconfigured Field Mappings for Business Component - CUT Address

Table 39 shows the data cleansing field mappings for the CUT Address business component and data cleansing operation.

NOTE: For Siebel Industry Applications, the CUT Address business component is used instead of the Business Address business component.

Table 39. Preconfigured Field Mappings for ODQ Address Validation Server Business Component - CUT Address

Business Component Field	Mapped Field
City	Business Address.City
Country	Business Address.Country
Postal Code	Business Address.Postal Code
State	Business Address.State
Street Address	Business Address.Street Address
Street Address 2	Business Address.Street Address 2

Preconfigured Field Mappings for Business Component - Personal Address

Table 40 shows the data cleansing field mappings for the Personal Address business component and data cleansing operation.

Table 40. Preconfigured Field Mappings for ODQ Address Validation Server Business Component - Personal Address

Business Component Field	Mapped Field
City	Business Address.City
Country	Business Address.Country
Postal Code	Business Address.Postal Code
State	Business Address.State
Street Address	Business Address.Street Address
Street Address 2	Business Address.Street Address 2

C Preconfigured Parameter and Field Mapping Values for SDQ Matching Server

This appendix provides examples of the preconfigured parameter and field mapping values for the SDQ Matching Server. The SDQ Matching Server is preconfigured for Search Software America (SSA) for data matching.

This appendix includes the following topics:

- About Parameter and Field Mapping Values for SDQ Matching Server on page 143
- Preconfigured Vendor Parameters for SSA on page 144
- Preconfigured Field Mappings for SSA on page 149

About Parameter and Field Mapping Values for SDQ Matching Server

Matching Server definitions are configured as vendor parameters in the Administration - Data Quality, then Third Party Administration view. Use the following procedure to access and view the preconfigured vendor parameters. For more information about vendor parameter configuration, see "Configuring Vendor Parameters" on page 57.

To view the preconfigured vendor parameters for SSA

1. Navigate to the Administration - Data Quality screen, then Third Party Administration view.
2. In the Vendor list, select the record with the name SSA.
3. Click the Vendor Parameter view tab.

   The vendor parameters are displayed in the Vendor Parameters list.

The field mappings from vendor fields to Siebel application fields are configured in field mapping parameters in the Administration - Data Quality screen, Third Party Administration view. There are field mappings for each of the supported business components and operations. Use the following procedure to view the preconfigured field mappings for SSA applications. For information about mapping fields for data matching, see "Mapping of Vendor Fields to Business Component Fields" on page 57.

To view preconfigured field mappings for SSA

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

   The field mappings are displayed in the Field Mapping list.

Preconfigured Vendor Parameters for SSA

Table 41 lists the preconfigured vendor parameters for SSA.

Table 41. Preconfigured Vendor Properties for SSA

Name	Value	Configurable	Description
Account DeDup Record Type	Account	Yes	Record type for the Account, Business Address, Contact, and List Mgmt Prospective Contact business components.
Business Address DeDup Record Type	Business Address	Yes	
Contact DeDup Record Type	Contact	Yes	
List Mgmt Prospective Contact DeDup Record Type	List Mgmt Prospective Contact	Yes	
Personal Address DeDup Record Type	Business Address	Yes	Merge cell with previous cell.
SSA Account Match Purpose	Company_Optional	Yes	Match purpose for Account, Contact, and List Mgmt Prospective Contact. For more information about match purpose, see "Match Purpose" on page 67.
SSA Contact Match Purpose	Company_Optional	Yes	
SSA List Mgmt Prospective Contact Match Purpose	Contact_Optional	Yes	
SSA Calculated Fields 1	"First", "N", "0"	No	Calculated vendor fields
SSA Calculated Fields 2	"Middle", "N", "1"	No	Calculated vendor fields
SSA Calculated Fields 3	"Last", "N", "2"	No	Calculated vendor fields
SSA Calculated Fields 4	"City", "B", "0"	No	Calculated vendor fields
SSA Calculated Fields 5	"State", "B", "1"	No	Calculated vendor fields
SSA Calculated Fields 6	"Country", "B", "2"	No	Calculated vendor fields

Table 41. Preconfigured Vendor Properties for SSA

Name	Value	Configurable	Description
SSA Default Universal Setting Key Type	Standard	Yes. Select the key type during installation and do not change after keys are generated.	Specifies the numbers of match keys to be generated: ■ **Limited**. For overcoming less sequence variation; fewer keys are generated ■ **Standard.** For overcoming more sequence variation; more keys are generated. For more information about key type, see Table 13 on page 43 and "Match Key Generation with the Matching Server" on page 23.
SSA Default Universal Setting Match Level	Typical	Yes	Specifies how closely matched the records must be before they are accepted as a match: ■ **Conservative**. Definite matches ■ **Typical**. Possible matches ■ **Loose**. Least likely matches
SSA Default Universal Setting Search Type	Typical	Yes	Controls the number of candidate records retrieved by the search ranges: For more information about search type, see "Identification of Candidate Records with the Universal Connector" on page 24.

Table 41. Preconfigured Vendor Properties for SSA

Name	Value	Configurable	Description
SSA Default Universal Setting Threshold	75	Yes	A score threshold that determines a match decision as either Accept, Undecided, or Reject. For more information about match thresholds, see Table 13 on page 43.
SSA Invalid Param Error Code	"-1", "Population", "-2", "Code Page", "-3", "ABC", "-4", "Key Type", "-7", "Search Type", "-8", "Match Purpose", "-9", "Match Level"	No	The error codes that indicate that a passed parameter is invalid. For example, in '"-1", "Population"', the error code -1 indicates the parameter is invalid for Population.
SSA Key Field	"Contact_Optional", "Person", "Contact_Mandatory", "Person", "Company_Mandatory", "Company", "Company_Optional", "Company"	No	The mapping of fields for the Match Purpose parameter. For example, the Siebel CRM name, Contact_Optional, is mapped to the SSA Match Purpose name of Person. The application sends Person to SSA as a Match Purpose parameter if the user selects the match purpose Contact_Optional from the application.
SSA Max Combined Query Records	50	Yes	The maximum number of combined query ranges for a match record. Used by SDQ code. **NOTE:** This parameter is for batch mode only.
SSA Max Combined Ranges	20	Yes	The maximum number of combined ranges for the matcher cache. Used by SDQ code. **NOTE:** This parameter is for batch mode only.

Table 41. Preconfigured Vendor Properties for SSA

Name	Value	Configurable	Description
SSA Max Matches	0	Yes	The maximum number of match candidate records to process at a time. A value of 0 indicates that all data will be processed at the same time. Used by SDQ code. **NOTE:** This parameter operates in real-time or in batch mode.
SSA Supported Fields 1	"Company_Mandatory", "C", "C", "A", "A", "I", "I", "Z", "Z", "B", "B", "O", "C", "L", "B", "S", "A"	No	Field types used for the match purpose Company_Mandatory
SSA Supported Fields 2	"Company_Optional", "C", "C", "S", "S", "I", "I", "Z", "Z", "L", "L", "B", "L", "A", "S"	No	Field types used for the match purpose Company_Optional
SSA Supported Fields 3	"Contact_Mandatory", "N", "N", "E", "E", "T", "T", "C", "C", "A", "A", "I", "I", "Z", "Z", "B", "B", "O", "C", "L", "B", "S", "A", "D", "D"	No	Field types used for the match purpose Contact_Mandatory
SSA Supported Fields 4	"Contact_Optional", "N", "N", "E", "E", "T", "T", "O", "O", "S", "S", "I", "I", "Z", "Z", "L", "L", "C", "O", "B", "L", "A", "S", "D", "D"	No	Field types used for the match purpose Contact_Optional
SSA Vendor Key Field	"Contact_Optional", "N", "Contact_Mandatory", "N", "Company_Mandatory", "C", "Company_Optional", "C"	No	Mapping of match purpose to the vendor field name.
SSA Population-Codepage ARA	"Arabic", "Arabic"	No	Specifies the country and code page.
SSA Population-Codepage CHS	"China", "Chinese_Simp"	No	
SSA Population-Codepage CHT	"China", "Chinese_Trad"	No	
SSA Population-Codepage CSY	"Czech","Latin_2_1250"	No	

Table 41. Preconfigured Vendor Properties for SSA

Name	Value	Configurable	Description
SSA Population-Codepage DAN	"Denmark", "Latin_1_Mixed"	No	Specifies the country and code page.
SSA Population-Codepage DEU	"Germany", "Latin_1_Mixed"	No	
SSA Population-Codepage ELL	"Greece", "Greek"	No	
SSA Population-Codepage ENU	"Default", "Latin_1_Mixed"	No	
SSA Population-Codepage ESN	"Spain", "Latin_1_Mixed"	No	
SSA Population-Codepage FIN	"Finland", "Latin_1_Mixed"	No	
SSA Population-Codepage FRA	"France", "Latin_1_Mixed"	No	
SSA Population-Codepage HEB	"Israel", "Hebrew"	No	
SSA Population-Codepage ITA	"Italy", "Latin_1_Mixed"	No	
SSA Population-Codepage JPN	"Japan", "Japanese"	No	
SSA Population-Codepage KOR	"South_Korea", "Korean"	No	
SSA Population-Codepage NLD	"Netherlands", "Latin_1_Mixed"	No	
SSA Population-Codepage PLK	"Poland", "Latin_2_1250"	No	
SSA Population-Codepage PSL	"Default", "Latin_1_Mixed"	No	
SSA Population-Codepage PTB	"Brazil", Latin_1_Mixed"	No	
SSA Population-Codepage PTG	"Portugal", "Latin_1_Mixed"	No	
SSA Population-Codepage SVE	"Sweden", "Latin_1_Mixed"	No	
SSA Population-Codepage THA	"Thailand", "Thai"	No	

Preconfigured Field Mappings for SSA

This topic lists the preconfigured field mappings for the various business components and operations supported by SSA applications, including:

- "Preconfigured Field Mappings for Business Component- Account" on page 149
- "Preconfigured Field Mappings for Business Component - Contact" on page 150
- "Preconfigured Field Mappings for Business Component - List Mgmt Prospective Contact" on page 151
- "Preconfigured Field Mappings for Business Component - Business Address" on page 152

Some of the mapped 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 relevant documentation included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery.

NOTE: The tables in this section indicate when field mappings are different for Siebel Industry Applications.

Preconfigured Field Mappings for Business Component- Account

Table 42 shows the data matching field mappings for the Account business component and DeDuplication operation.

Table 42. Preconfigured SSA Data Matching Mappings for the Account Business Component

Business Component Field	Mapped Field
Primary Account City	City
Primary Account Country	Country
DUNS Number	I
Name	C
Primary Account Postal Code	Z
Primary Account State	State
Primary Account Street Address	A

Preconfigured Field Mappings for Business Component - Contact

Table 43 shows the data matching field mappings for the Contact business component and DeDuplication operation.

Table 43. Preconfigured SSA Data Matching Mappings for the Contact Business Component

Business Component Field	Mapped Field
Primary Account Name Account (Siebel Industry Applications)	C
Birth Date	D
Cellular Phone #	T
Primary City Primary Personal City (Siebel Industry Applications)	City
Primary Country Primary Personal Country (Siebel Industry Applications)	Country
Email Address	E
First Name	First
Home Phone #	T
Last Name	Last
Middle Name	Middle
Primary Postal Code Primary Personal Postal Code (Siebel Industry Applications)	Z
Social Security Number	I
Primary State Primary Personal State (Siebel Industry Applications)	State
Primary Street Address Primary Personal Street Address (Siebel Industry Applications)	A
Work Phone #	T

Preconfigured Field Mappings for Business Component - List Mgmt Prospective Contact

Table 44 shows the data matching field mappings for the List Mgmt Prospective Contact business component and DeDuplication operation.

Table 44. Preconfigured SSA Data Matching Mappings for the List Mgmt Prospective Contact Business Component

Business Component Field	Mapped Field
Account Name Account (Siebel Industry Applications)	C
Cellular Phone #	T
City	City
Country	Country
Email Address	E
First Name	First
Home Phone #	T
Last Name	Last
Middle Name	Middle
Postal Code	Z
Social Security Number	I
State	State
Street Address	A
Work Phone #	T

Preconfigured Field Mappings for Business Component - Business Address

Table 45 shows the data matching field mappings for the Business Address business component and DeDuplication operation.

For Siebel Industry Applications, the CUT Address business component is used instead of the Business Address business component.

NOTE: This mapping is required to support the automatic deduplication on address update functionality.

Table 45. Preconfigured SSA Data Matching Mappings for the Business Address Business Component

Business Component Field	Mapped Field
City	City
Country	Country
Id	Id
Postal Code	Z
State	State
Street Address	A

D 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) or shared libraries that they provide for use with the SDQ Universal Connector. It includes the following topics:

Vendor Libraries

Vendors must follow these rules for their DLLs or shared libraries:

- The libraries must be thread-safe. A library can support multiple sessions by using different unique session IDs.
- The libraries must support UTF-16 (UCS2) as the default Unicode encoding.
- If there is a single library for all supported languages, the libraries must be named as follows:
  - *BASE*.dll (on Windows)
  - lib*BASE*.so (on AIX and Oracle Solaris)
  - lib*BASE*.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 libraries.
- If there are separate libraries for different languages, the library name must include the appropriate language code. For example, for Japanese (JPN), the libraries must be named as follows:
  - *BASE*jpn.dll (on Windows)
  - lib*BASE*jpn.so (on AIX and Oracle Solaris)
  - lib*BASE*jpn.sl (on HP-UX)

The Siebel application loads the libraries from the locations described in Table 11 on page 36.

The mapping of Siebel application field names to vendor field names is stored as values of the relevant Business Component user properties in the Siebel CRM repository. Storage of these field values is mandatory.

Any other vendor-specific parameter required (for example, port number) for the vendor's library must be stored outside of Siebel CRM.

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 the functions that are called when the vendor library is loaded or when the Siebel Server shuts down:

- "sdq_init_connector Function" on page 154
- "sdq_shutdown_connector Function" on page 155

sdq_init_connector Function

This function is called using the absolute installation path of the SDQConnector directory (.\siebsrvr\SDQConnector) when the vendor library is first loaded to facilitate any initialization tasks. It can be used by the vendor to read any configuration files it might choose to use.

Syntax	`int sdq_init_connector (const SSchar * path)`
Parameters	path: The absolute path of the Siebel Server installation. Vendors can use this path to locate any required parameter file for loading the necessary parameters (like port number and so on). This is a Unicode string because the Siebel Server can be installed for languages other than English.
Return Value	A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function.

sdq_shutdown_connector Function

This function is called when the Siebel Server is shutting down to perform any necessary cleanup tasks.

Syntax `int sdq_shutdown_connector (void)`

Parameters This function does not have any parameters.

Return Value A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function.

Session Initialization and Termination Functions

The Siebel Server works in multi-threaded mode to serve multiple users. To allow for user and invocation-specific parameters, there is the concept of a session context where such values can be stored. The session ID is supplied for all data matching or data cleansing functions. Upon completion of data cleansing or data matching the session is closed.

This topic describes the functions that are used for session initialization and termination:

- "sdq_init_session Function" on page 155
- "sdq_close_session Function" on page 156

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 session_id: A unique value provided by the vendor that is used in function calls while the session is active. The value 0 is reserved as an invalid session ID. The Siebel CRM code calls this function with a session ID of 0, so the session ID must be initialized to a nonzero value.

Return Value A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function.

sdq_close_session Function

This function is called when a particular data cleansing or data matching operation is finished and it is required to close the session. Any necessary cleanup tasks are performed.

Syntax	`int sdq_close_session (int * session_id)`
Parameters	session_id: The session ID obtained by initializing the session.
Return Value	A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function.

Parameter Setting Functions

Most third party software vendors provide lists of parameters to customers so that the customers can configure the vendor library's behavior to suit their business needs.

This topic describes the functions that set parameters at both the global context and at the session context (that is, specific to a session).

- "sdq_set_global_parameter Function" on page 156
- "sdq_set_parameter Function" on page 157

sdq_set_global_parameter Function

This function is called to set global parameters. The function call is made after the call to sdq_init_connector. The vendor must put the configuration file, if using one, in `.\siebsrvr\SDQConnectorpath`. When the vendor DLL is loaded, it calls the sdq_init_connector API function (if it is exposed by the vendor) with the absolute path to the SDQConnector directory. It is then up to the vendor to read the appropriate configuration file. The configuration file name is dependent on vendor specifications.

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

Using the sdq_set_global_parameter API, any global parameters specific to the vendor can be put as a user property to DeDuplication business service, where the format of the business service user property is as follows:

```
"Global", "Parameter Name", "Parameter Value"
```

These global parameters are set to the vendor only after the vendor DLL loads. You can define user properties for the DeDuplication business service as follows:

Property: My Connector 1
Value: MyDQMatch

Property: MyDQMatch Parameter 1
Value: "Global", "zGlobalParam1", "zGlobalParam1Val"

NOTE: 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 parameterList: An XML character string that contains the list of parameters and values specific to this function call. An example of the XML is as follows:

```
<Data>
  <Parameter>
     <GlobalParam1>GlobalParam1Val</GlobalParam1>
  </Parameter>
</Data>
```

Return Value A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function.

sdq_set_parameter Function

This function is called, after the call to sdq_init_session, to set parameters that are applicable at the session context. The vendor must put the configuration file, if using one, in `.\siebsrvr\SDQConnectorpath`. When the vendor DLL is loaded, it calls the sdq_init_connector API function (if it is exposed by the vendor) with the absolute path to the SDQConnector directory. It is then up to the vendor to read the appropriate configuration file. The configuration file name is dependent on vendor specifications.

Using the sdq_set_parameter API, any session parameters specific to the vendor can be put as a user property to the DeDuplication business service, where the format of the business service user property is as follows:

```
"Session", “Parameter Name”, “Parameter Value”
```

These session parameters are set to the vendor, after each session opens with the vendor. Your can define user properties for the DeDuplication business service as follows:

Property: My Connector 1
Value: MyDQMatch

Property: MyDQMatch Parameter 2
Value: "Session", "zSessParam2", "zSessParam2Val"

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

Parameters

- session_id: The session ID obtained while initializing the session.
- parameterList: An XML character string that contains the list of parameters and values that are specific to this function call. An example of the XML is as follows:

```
<Data>
 <Parameter>
    <Name>RECORD_TYPE</Name>
    <Value>Contact</Value>
 </Parameter>
```

```
    <Parameter>
      <Name>SessionParam1</Name>
      <Value>SessionValue1</Value>
    </Parameter>
  </Data>
```

Return Value A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function.

Error Message Functions

This topic describes the function that is associated with error messages: "sdq_get_error_message Function" on page 158.

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

- 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 different functions that are called for real-time data matching when match candidate acquisition takes place in Siebel CRM and in ODQ Matching Server.

- "sdq_dedup_realtime Function" on page 158 is used when match candidate acquisition takes place in Siebel CRM.
- "sdq_dedup_realtime_nomemory Function" on page 160 is used when match candidate acquisition takes place in ODQ Matching Server.

sdq_dedup_realtime Function

This function is called to perform real-time data matching when match candidate acquisition takes place in Siebel CRM.

This function sends the data for each record as driver records and their candidate records. The function is called only once; multiple calls to the vendor library are not made even when the set of potential candidate records is huge. As all the candidate records are sent at once, all the duplicates for a given record are returned.

Syntax

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

Parameters

- session_id: The session ID obtained by initializing the session.
- parameterList: An XML character string that contains the list of parameters and values that are specific to this function call. An XML example follows:

```
<Data>
 <Parameter>
   <Name>RealTimeDedupParam1</Name>
   <Value>RealTimeDedupValue1</Value>
 </Parameter>

 <Parameter>
   <Name>RealTimeDedupParam2</Name>
   <Value>RealTimeDedupValue2</Value>
 </Parameter>
</Data>
```

  NOTE: The parameterList parameter is set to NULL as all required parameters are already set at the session level.

- inputRecordSet: An XML character string containing the driver record and candidate records. An XML example follows:

```
<Data>
 <DriverRecord>
   <Account.Id>1-X42</Account.Id>
   <Account.Name>Siebel</Account.Name>
   <Account.Location>Headquarters</Account.Location>
 </DriverRecord>

 <CandidateRecord>
   <Account.Id>1-Y28</Account.Id>
   <Account.Name>Siebel</Account.Name>
   <Account.Location>Atlanta</Account.Location>
 </CandidateRecord>

 <CandidateRecord>
   <Account.Id>1-3-P</Account.Id>
   <Account.Name>Siebel</Account.Name>
   <Account.Location>Rome</Account.Location>
 </CandidateRecord>
</Data>
```

- outputRecordSet: An XML character string populated by the vendor in real time that contains the duplicate records with the scores. An XML example follows:

```
<Data>
 <DuplicateRecord>
   <Account.Id>SAME ID AS DRIVER </Account.Id>
   <DQ.MatchScore></DQ.MatchScore>
 </DuplicateRecord>

 <DuplicateRecord>
   <Account.Id>1-Y28</Account.Id>
   <DQ.MatchScore>92</DQ.MatchScore>
 </DuplicateRecord>

 <DuplicateRecord>
   <Account.Id>1-3-P</Account.Id>
   <DQ.MatchScore>88</DQ.MatchScore>
 </DuplicateRecord>
</Data>
```

Return Value A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function.

sdq_dedup_realtime_nomemory Function

This function is called to perform real-time data matching when match candidate acquisition takes place in ODQ Matching Server.

Syntax

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

Parameters session_id: The session ID obtained by initializing the session.

- parameterList: An XML character string that contains the list of parameters and values that are specific to this function call. An XML example follows:

```
<Data>
 <Parameter>
   <Name>RealTimeDedupParam1</Name>
   <Value>RealTimeDedupValue1</Value>
 </Parameter>

 <Parameter>
   <Name>RealTimeDedupParam2</Name>
   <Value>RealTimeDedupValue2</Value>
 </Parameter>
</Data>
```

NOTE: The parameterList parameter is set to NULL as all required parameters are already set at the session level.

- inputRecordSet: An XML character string containing the driver record. An XML example follows:

```
<Data>
 <DriverRecord>
    <DUNSNumber>123456789</DUNSNumber>
    <Name>Siebel</Name>
    <<RowId>1-X40</RowId>
 </DriverRecord>
</Data>
```

- outputRecordSet: An XML character string populated by the vendor in real time that contains the duplicate records with the scores. An XML example follows:

```
<Data>
 <DuplicateRecord>
    <Account.Id>SAME ID AS DRIVER </Account.Id>
    <DQ.MatchScore></DQ.MatchScore>
 </DuplicateRecord>

 <DuplicateRecord>
    <Account.Id>1-Y28</Account.Id>
    <DQ.MatchScore>92</DQ.MatchScore>
 </DuplicateRecord>

 <DuplicateRecord>
    <Account.Id>1-3-P</Account.Id>
    <DQ.MatchScore>88</DQ.MatchScore>
 </DuplicateRecord>
</Data>
```

Return Value A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function.

Batch Mode Data Matching Functions

This topic describes the functions that are called for batch mode data matching:

- "sdq_set_dedup_candidates Function" on page 162
- "sdq_start_dedup Function" on page 164
- "sdq_get_duplicates Function" on page 165

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

- session_id: The session ID obtained by initializing the session.
- parameterList: An XML character string that contains the list of parameters and values that are specific to this function call. An example of the XML is as follows:

  ```
  <Data>
   <Parameter>
     <Name>BatchDedupParam1</Name>
     <Value>BatchDedupValue1</Value>
   </Parameter>

   <Parameter>
     <Name>BatchDedupParam2</Name>
     <Value>BatchDedupValue2</Value>
   </Parameter>
  </Data>
  ```

  NOTE: The parameterList parameter is set to NULL as all required parameters are already set at the session level.

- xmlRecordSet: When match candidate acquisition takes place in Siebel CRM, the xmlRecordSet parameter is used as follows:
  - For full data matching batch jobs: An XML character string containing a list of candidate records. There is no driver record in the input set. An example of the XML is as follows:

    ```
    <Data>
      <CandidateRecord>
        <Account.Id>2-24-E</Account.Id>
        <Account.Name>Siebel</Account.Name>
        <Account.Location>Somewhere</Account.Location>
      </CandidateRecord>
      <CandidateRecord>
        <Account.Id>1-E-2E</Account.Id>
        <Account.Name>Siebel</Account.Name>
        <Account.Location>Somewhere else</Account.Location>
      </CandidateRecord>
      <CandidateRecord>
        <Account.Id>2-34-F</Account.Id>
        <Account.Name>Siebel</Account.Name>
        <Account.Location>Someplace</Account.Location>
      </CandidateRecord>
    </Data>
    ```

- For incremental data matching batch jobs: As more candidate records are queried from the Siebel CRM database and sent to the vendor software, the driver records must be marked so that the vendor software knows which records must return duplicate records:

```
<Data>
   <DriverRecord>
      <Account.Id>2-24-E</Account.Id>
      <Account.Name>Siebel</Account.Name>
      <Account.Location>Somewhere</Account.Location>
   </DriverRecord>
   <CandidateRecord>
      <Account.Id>1-E-9E</Account.Id>
      <Account.Name>Siebel</Account.Name>
      <Account.Location>Somewhere else</Account.Location>
   </CandidateRecord>
   <DriverRecord>
      <Account.Id>1-E-2E</Account.Id>
      <Account.Name>Siebel</Account.Name>
      <Account.Location>Somewhere else</Account.Location>
   </DriverRecord>
   <CandidateRecord>
   <Account.Id>1-12-2H</Account.Id>
      <Account.Name>Siebel</Account.Name>
      <Account.Location>Somewhere else</Account.Location>
   </CandidateRecord>
   <DriverRecord>
      <Account.Id>2-34-F</Account.Id>
      <Account.Name>Siebel</Account.Name>
      <Account.Location>Someplace</Account.Location>
   </DriverRecord>
</Data>
```

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.

- xmlRecordSet: When match candidate acquisition takes place in ODQ Matching Server, the xmlRecordSet parameter is used as follows:
  - For full data matching batch jobs, an empty string is sent.
  - For incremental data matching batch jobs, only driver records are sent.

  An example of the XML is as follows:

```
<Data>
  <DriverRecord>
      <DUNSNumber>123456789</DUNSNumber>
      <Name>Siebel</Name>
      <RowId>1-X40</RowId>
  </DriverRecord>
  <DriverRecord>
      <DUNSNumber>987654321</DUNSNumber>
      <Name>Oracle</Name>
      <RowId>1-X50</RowId>
  </DriverRecord>
  <DriverRecord>
      <DUNSNumber>123123123</DUNSNumber>
      <Name>IBM</Name>
      <RowId>1-X60</RowId>
  </DriverRecord>
</Data>
```

Return Value A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function.

sdq_start_dedup Function

This function is called to start the data matching process in batch mode, and essentially signals that all the records to be used for data matching have been sent to the vendor's application.

Syntax `int sdq_start_dedup (int session_id)`

Parameters session_id: The session ID obtained by initializing the session.

Return Value This function does not have a return value.

sdq_get_duplicates Function

This function is called to get the 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 for each 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, SSchar* xmlResultSet)`

Parameters session_id: The session ID obtained by initializing the session.

xmlRecordSet : An XML character string that the vendor library populates with a master record and a list of its duplicate records along with their match scores.

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:

```
<Data>
  <ParentRecord>
     <DQ.MasterRecordsRowID>2-24-E</DQ.MasterRecordsRowID>
       <DuplicateRecord>
           <Account.Id>2-24-E</Account.Id>
          <DQ.MatchScore>92</DQ.MatchScore>
       </DuplicateRecord>

       <DuplicateRecord>
          <Account.Id>2-23-F</Account.Id>
          <DQ.MatchScore>88</DQ.MatchScore>
       </DuplicateRecord>
  </ParentRecord>
</Data>
```

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 function that is called for real-time data matching: "sdq_datacleanse Function" on page 166.

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

- parameterList: An XML character string that contains the list of parameters and values that are specific to this function call. An example of the XML is as follows:

  ```
  <Data>
      <Parameter>
         <Name>RealTimeDataCleanseParam1</Name>
         <Value>RealTimeDataCleanseValue1</Value>
      </Parameter>

      <Parameter>
         <Name>RealTimeDataCleanseParam2</Name>
         <Value>RealTimeDataCleanseValue2</Value>
      </Parameter>
  </Data>
  ```

  NOTE: This parameter is set to NULL as all required parameters are already set at the session level.

- inputRecordSet: An XML character string containing the driver record. An example of the XML is as follows:

  ```
  <Data>
      <DriverRecord>
         <Contact.FirstName>michael</Contact.FirstName>
         <Contact.LastName>mouse</Contact.LastName>
      </DriverRecord>
  </Data>
  ```

- 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:

  ```
  <Data>
      <CleansedDriverRecord>
         <Contact.FirstName>Michael</Contact.FirstName>
         <Contact.LastName>Mouse</Contact.LastName>
      </CleansedDriverRecord>
  </Data>
  ```

Return Value A return value of 0 indicates successful execution. Any other value is a vendor error code. The error message details from the vendor are obtained by calling the sdq_get_error_message function.

Batch Mode Data Cleansing Functions

This topic describes the function that is called for batch mode data cleansing: "sdq_data_cleanse Function" on page 167.

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 following algorithms used in Siebel code for both real-time and batch data cleansing and data matching:

- "Batch Data Matching Algorithm" on page 167
- "Real-Time Data Matching Algorithm" on page 168
- "Batch Data Cleansing Algorithm" on page 168
- "Real-Time Data Cleansing Algorithm" on page 169

Batch Data Matching Algorithm

The algorithm is as follows:

1. Load the vendor library.
2. Call sdq_init_connector.
3. Call sdq_set_global_parameter.
4. Call sdq_init_session.
5. Call sdq_set_parameter (RECORD_TYPE – Account/Contact/List Mgmt Prospective Contact, BATCH_DATAFLOW_NAME, BATCH_MATCH_MAX_NUM_OF_RECORDS)
6. Query the Siebel CRM 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. Siebel Data Quality Universal Connector supports multiple key generation.

   For more information about match key generation, see "Generating or Refreshing Keys Using Batch Jobs" on page 79.
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 library.

2 Call sdq_init_connector.

3 Call sdq_set_global_parameter.

4 Call sdq_init_session.

5 Call sdq_set_parameter (RECORD_TYPE – Account/Contact/List Mgmt Prospective Contact).

6 Query the Siebel CRM 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 library.

2 Call sdq_init_connector.

3 Call sdq_set_global_parameter.

4 Call sdq_init_session.

5 Call sdq_set_parameter (RECORD_TYPE – Account/Business Address/Contact/List Mgmt Prospective Contact, BATCH_DATAFLOW_NAME).

6 Query the Siebel CRM 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 CRM repository.

9 Call sdq_close_session while logging out of current session.

10 Call sdq_close_connector.

Real-Time Data Cleansing Algorithm

The algorithm is as follows:

1. Load the vendor library.
2. Call sdq_init_connector.
3. Call sdq_set_global_parameter.
4. Call sdq_init_session.
5. Call sdq_set_parameter (RECORD_TYPE – Account/Business Address/Contact/List Mgmt Prospective Contact).
6. Query the Siebel CRM 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 CRM repository.
9. Call sdq_close_session while logging out of current session
10. Call sdq_close_connector.

E Siebel Business Applications Action Sets

This appendix introduces the Siebel Business Applications action sets that are set up by default in your Siebel application for Account, Contact and List Mgmt Prospective Contact. It includes the following topics:

- Setting up Siebel Business Application Action Sets Manually on page 171
- Siebel Business Applications Action Sets for Account on page 171
- Siebel Business Applications Action Sets for Contact on page 178
- Siebel Business Applications Action Sets for List Mgmt Prospective Contact on page 185
- Siebel Business Applications Generic Action Sets on page 192

Setting up Siebel Business Application Action Sets Manually

Use the following procedure to manually set up additional, non default Siebel application action sets.

To manually set up additional Siebel application action sets

- Navigate to the Administration - Runtime Events screen, then the Action Sets view.

Use the following procedure to manually set up additional Siebel application run-time events.

To manually set up additional Siebel application run-time events

- Navigate to the Administration Runtime Evens screen, then the Events view.

Siebel Business Applications Action Sets for Account

This topic introduces the following Siebel application action sets for Account:

- "ISSLoad Account" on page 172
- "ISSSYNC DeleteRecord Account" on page 173
- "ISSSYNC PreDeleteRecord Account" on page 174
- "ISSSYNC PreWriteRecord Account" on page 175
- "ISSSYNC WriteRecord Account" on page 176

For more information about creating action sets, including creating actions for action sets, and associating events with action sets, see *Siebel Personalization Administration Guide*.

ISSLoad Account

Table 46 describes the actions in the ISSLoad Account action set.

Table 46. Actions in ISSLoad Account Action Set

Action	Name of Field	Value
ISS Set System Name	Name	ISS Set System Name
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_SYSTEM
	Set Operator	Set
	Value	SiebelDQ
ISS Set Page Size	Name	ISS Set Page Size
	Sequence	2
	Action Type	Attribute Set
	Profile Attribute	IDS_PAGE_SIZE
	Set Operator	Set
	Value	80
ISS Set File Name	Name	ISS Set File Name
	Sequence	3
	Action Type	Attribute Set
	Profile Attribute	IDS_LOADFILE
	Set Operator	Set
	Value	"C:\ids\iss2704s\ids\data\account.xml" **NOTE:** Modify this value if you install ODQ Matching Server on a drive other than `C:\` drive.
ISS Set IDT Name	Name	ISS Set IDT Name
	Sequence	4
	Action Type	Attribute Set
	Profile Attribute	IDS_IDT
	Set Operator	Set
	Value	IDS_01_IDT_ACCOUNT

Table 46. Actions in ISSLoad Account Action Set

Action	Name of Field	Value
ISS Set IO Name	Name	ISS Set IO Name
	Sequence	5
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_NAME
	Set Operator	Set
	Value	ISS_Account
ISS Run WF	Name	ISS Run WF
	Sequence	6
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch Build Load File"

ISSSYNC DeleteRecord Account

Table 47 describes the actions in the ISSSYNC DeleteRecord Account action set.

Table 47. Actions in ISSSYNC DeleteRecord Account Action Set

Action	Name of Field	Value
ISS Set URL	Name	ISS Set URL
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_URL
	Set Operator	Set
	Value	"http://SERVERNAME:1671" **NOTE:** Replace SERVERNAME with the Hostname or IP address of the computer where XML Sync Server is installed.

Table 47. Actions in ISSSYNC DeleteRecord Account Action Set

Action	Name of Field	Value
ISS Run WF	Name	ISS Run WF
	Sequence	2
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch Delete Record Sync"

ISSSYNC PreDeleteRecord Account

Table 48 describes the actions in the ISSSYNC PreDeleteRecord Account action set.

Table 48. Actions in ISSSYNC PreDeleteRecord Account Action Set

Action	Name of Field	Value
ISS Set System Name	Name	ISS Set System Name
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_SYSTEM
	Set Operator	Set
	Value	SiebelDQ
ISS Set IDT Name	Name	ISS Set IDT Name
	Sequence	2
	Action Type	Attribute Set
	Profile Attribute	IDS_IDT
	Set Operator	Set
	Value	IDS_01_IDT_ACCOUNT
ISS Set IO Name	Name	ISS Set IO Name
	Sequence	3
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_NAME
	Set Operator	Set
	Value	ISS_Account

Table 48. Actions in ISSSYNC PreDeleteRecord Account Action Set

Action	Name of Field	Value
ISS Set ID	Name	ISS Set ID
	Sequence	4
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_ID
	Set Operator	Set
	Value	[Id]
ISS Run WF	Name	ISS Run WF
	Sequence	5
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch PreDelete Record Sync"

ISSSYNC PreWriteRecord Account

Table 49 describes the actions in the ISSSYNC PreWriteRecord Account action set.

Table 49. Actions in ISSSYNC PreWriteRecord Account Action Set

Action	Name of Field	Value
ISS Set System Name	Name	ISS Set System Name
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_SYSTEM
	Set Operator	Set
	Value	SiebelDQ
ISS Set IDT Name	Name	ISS Set IDT Name
	Sequence	2
	Action Type	Attribute Set
	Profile Attribute	IDS_IDT
	Set Operator	Set
	Value	IDS_01_IDT_ACCOUNT

Table 49. Actions in ISSSYNC PreWriteRecord Account Action Set

Action	Name of Field	Value
ISS Set IO Name	Name	ISS Set IO Name
	Sequence	3
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_NAME
	Set Operator	Set
	Value	ISS_Account
ISS Set ID	Name	ISS Set ID
	Sequence	4
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_ID
	Set Operator	Set
	Value	[Id]
ISS Run WF	Name	ISS Run WF
	Sequence	5
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch PreWrite Record Sync"

ISSSYNC WriteRecord Account

Table 50 describes the actions in the ISSYNC WriteRecord Account action set.

Table 50. Actions in ISSSYNC WriteRecord Account Action Set

Action	Name of Field	Value
ISS Set System Name	Name	ISS Set System Name
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_SYSTEM
	Set Operator	Set
	Value	SiebelDQ

Table 50. Actions in ISSSYNC WriteRecord Account Action Set

Action	Name of Field	Value
ISS Set IDT Name	Name	ISS Set IDT Name
	Sequence	2
	Action Type	Attribute Set
	Profile Attribute	IDS_IDT
	Set Operator	Set
	Value	IDS_01_IDT_ACCOUNT
ISS Set IO Name	Name	ISS Set IO Name
	Sequence	3
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_NAME
	Set Operator	Set
	Value	ISS_Account
ISS Set ID	Name	ISS Set ID
	Sequence	4
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_ID
	Set Operator	Set
	Value	[Id]
ISS Set URL	Name	ISS Set URL
	Sequence	5
	Action Type	Attribute Set
	Profile Attribute	IDS_URL
	Set Operator	Set
	Value	"http://SERVERNAME:1671" **NOTE:** Replace SERVERNAME with the Hostname or IP address of the computer where XML Sync Server is installed.
ISS Run WF	Name	ISS Run WF
	Sequence	6
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch Write Record Sync"

Siebel Business Applications Action Sets for Contact

This topic introduces the following Siebel application action sets for Contact:

- "ISSLoad Contact" on page 178
- "ISSSYNC DeleteRecord Contact" on page 180
- "ISSSYNC PreDeleteRecord Contact" on page 180
- "ISSSYNC PreWriteRecord Contact" on page 182
- "ISSSYNC WriteRecord Contact" on page 183

For more information about creating action sets, including creating actions for action sets, and associating events with action sets, see *Siebel Personalization Administration Guide*.

ISSLoad Contact

Table 51 describes the actions in the ISSLoad Contact action set.

Table 51. Actions in ISSLoad Contact Action Set

Action	Name of Field	Value
ISS Set System Name	Name	ISS Set System Name
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_SYSTEM
	Set Operator	Set
	Value	SiebelDQ
ISS Set Page Size	Name	ISS Set Page Size
	Sequence	2
	Action Type	Attribute Set
	Profile Attribute	IDS_PAGE_SIZE
	Set Operator	Set
	Value	80

Table 51. Actions in ISSLoad Contact Action Set

Action	Name of Field	Value
ISS Set File Name	Name	ISS Set File Name
	Sequence	3
	Action Type	Attribute Set
	Profile Attribute	IDS_LOADFILE
	Set Operator	Set
	Value	"C:\ids\iss2704s\ids\data\contact.xml" **NOTE:** Modify this value if you install ODQ Matching Server on a drive other than `C:\` drive.
ISS Set IDT Name	Name	ISS Set IDT Name
	Sequence	4
	Action Type	Attribute Set
	Profile Attribute	IDS_IDT
	Set Operator	Set
	Value	IDS_01_IDT_CONTACT
ISS Set IO Name	Name	ISS Set IO Name
	Sequence	5
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_NAME
	Set Operator	Set
	Value	ISS_Contact
ISS Run WF	Name	ISS Run WF
	Sequence	6
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch Build Load File"

ISSSYNC DeleteRecord Contact

Table 52 describes the actions in the ISSSYNC DeleteRecord Contact action set.

Table 52. Actions in ISSSYNC DeleteRecord Contact Action Set

Action	Name of Field	Value
ISS Set URL	Name	ISS Set URL
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_URL
	Set Operator	Set
	Value	"http://SERVERNAME:1671" **NOTE:** Replace SERVERNAME with the Hostname or IP address of the computer where XML Sync Server is installed.
ISS Run WF	Name	ISS Run WF
	Sequence	2
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch Delete Record Sync"

ISSSYNC PreDeleteRecord Contact

Table 53 describes the actions in the ISSYNC PreDeleteRecord Contact action set.

Table 53. Actions in ISSSYNC PreDeleteRecord Contact Action Set

Action	Name of Field	Value
ISS Set System Name	Name	ISS Set System Name
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_SYSTEM
	Set Operator	Set
	Value	SiebelDQ

Table 53. Actions in ISSSYNC PreDeleteRecord Contact Action Set

Action	Name of Field	Value
ISS Set IDT Name	Name	ISS Set IDT Name
	Sequence	2
	Action Type	Attribute Set
	Profile Attribute	IDS_IDT
	Set Operator	Set
	Value	IDS_01_IDT_CONTACT
ISS Set IO Name	Name	ISS Set IO Name
	Sequence	3
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_NAME
	Set Operator	Set
	Value	ISS_Contact
ISS Set ID	Name	ISS Set ID
	Sequence	4
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_ID
	Set Operator	Set
	Value	[Id]
ISS Run WF	Name	ISS Run WF
	Sequence	5
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch PreDelete Record Sync"

ISSSYNC PreWriteRecord Contact

Table 54 describes the actions in the ISSSYNC PreWriteRecord Contact action set.

Table 54. Actions in ISSSYNC PreWriteRecord Contact Action Set

Action	Name of Field	Value
ISS Set System Name	Name	ISS Set System Name
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_SYSTEM
	Set Operator	Set
	Value	SiebelDQ
ISS Set IDT Name	Name	ISS Set IDT Name
	Sequence	2
	Action Type	Attribute Set
	Profile Attribute	IDS_IDT
	Set Operator	Set
	Value	IDS_01_IDT_CONTACT
ISS Set IO Name	Name	ISS Set IO Name
	Sequence	3
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_NAME
	Set Operator	Set
	Value	ISS_Contact
ISS Set ID	Name	ISS Set ID
	Sequence	4
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_ID
	Set Operator	Set
	Value	[Id]

Table 54. Actions in ISSSYNC PreWriteRecord Contact Action Set

Action	Name of Field	Value
ISS Run WF	Name	ISS Run WF
	Sequence	5
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch PreWrite Record Sync"

ISSSYNC WriteRecord Contact

Table 55 describes the actions in the ISSSYNC WriteRecord Contact action set.

Table 55. Actions in ISSSYNC WriteRecord Contact Action Set

Action	Name of Field	Value
ISS Set System Name	Name	ISS Set System Name
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_SYSTEM
	Set Operator	Set
	Value	SiebelDQ
ISS Set IDT Name	Name	ISS Set IDT Name
	Sequence	2
	Action Type	Attribute Set
	Profile Attribute	IDS_IDT
	Set Operator	Set
	Value	IDS_01_IDT_CONTACT
ISS Set IO Name	Name	ISS Set IO Name
	Sequence	3
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_NAME
	Set Operator	Set
	Value	ISS_Contact

Table 55. Actions in ISSSYNC WriteRecord Contact Action Set

Action	Name of Field	Value
ISS Set ID	Name	ISS Set ID
	Sequence	4
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_ID
	Set Operator	Set
	Value	[Id]
ISS Set URL	Name	ISS Set URL
	Sequence	5
	Action Type	Attribute Set
	Profile Attribute	IDS_URL
	Set Operator	Set
	Value	"http://SERVERNAME:1671" **NOTE:** Replace SERVERNAME with the Hostname or IP address of the computer where XML Sync Server is installed.
ISS Run WF	Name	ISS Run WF
	Sequence	6
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch Write Record Sync"

Siebel Business Applications Action Sets for List Mgmt Prospective Contact

This topic introduces the Siebel application action sets for List Mgmt Prospective Contact:

- "ISSLoad List Mgmt Prospective Contact" on page 185
- "ISSSYNC DeleteRecord List Mgmt Prospective Contact" on page 187
- "ISSSYNC PreDeleteRecord List Mgmt Prospective Contact" on page 188
- "ISSSYNC PreWriteRecord List Mgmt Prospective Contact" on page 189
- "ISSSYNC WriteRecord List Mgmt Prospective Contact" on page 190

For more information about creating action sets, including creating actions for action sets, and associating events with action sets, see *Siebel Personalization Administration Guide*.

ISSLoad List Mgmt Prospective Contact

Table 56 describes the actions in the ISSLoad List Mgmt Prospective Contact action set.

Table 56. Actions in ISSLoad List Mgmt Prospective Contact Action Set

Action	Name of Field	Value
ISS Set System Name	Name	ISS Set System Name
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_SYSTEM
	Set Operator	Set
	Value	SiebelDQ
ISS Set Page Size	Name	ISS Set Page Size
	Sequence	2
	Action Type	Attribute Set
	Profile Attribute	IDS_PAGE_SIZE
	Set Operator	Set
	Value	80

Table 56. Actions in ISSLoad List Mgmt Prospective Contact Action Set

Action	Name of Field	Value
ISS Set File Name	Name	ISS Set File Name
	Sequence	3
	Action Type	Attribute Set
	Profile Attribute	IDS_LOADFILE
	Set Operator	Set
	Value	"C:\ids\iss2704s\ids\data\prospect.xml" **NOTE:** Modify this value if you install ISS on a drive other than `C:\` drive.
ISS Set IDT Name	Name	ISS Set IDT Name
	Sequence	4
	Action Type	Attribute Set
	Profile Attribute	IDS_IDT
	Set Operator	Set
	Value	IDS_01_IDT_PROSPECT
ISS Set IO Name	Name	ISS Set IO Name
	Sequence	5
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_NAME
	Set Operator	Set
	Value	ISS_List_Mgmt_Prospective_Contact
ISS Run WF	Name	ISS Run WF
	Sequence	6
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch Build Load File"

ISSSYNC DeleteRecord List Mgmt Prospective Contact

Table 57 describes the actions in the ISSSYNC DeleteRecord List Mgmt Prospective Contact action set.

Table 57. Actions in ISSSYNC DeleteRecord List Mgmt Prospective Contact Action Set

Action	Name of Field	Value
ISS Set URL	Name	ISS Set URL
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_URL
	Set Operator	Set
	Value	"http://SERVERNAME:1671" **NOTE:** Replace SERVERNAME with the Hostname or IP address of the computer where XML Sync Server is installed.
ISS Run WF	Name	ISS Run WF
	Sequence	2
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch Delete Record Sync"

ISSSYNC PreDeleteRecord List Mgmt Prospective Contact

Table 58 describes the actions in the ISSSYNC PreDeleteRecord List Mgmt Prospective Contact action set.

Table 58. Actions in ISSSYNC PreDeleteRecord List Mgmt Prospective Contact Action Set

Action	Name of Field	Value
ISS Set System Name	Name	ISS Set System Name
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_SYSTEM
	Set Operator	Set
	Value	SiebelDQ
ISS Set IDT Name	Name	ISS Set IDT Name
	Sequence	2
	Action Type	Attribute Set
	Profile Attribute	IDS_IDT
	Set Operator	Set
	Value	IDS_01_IDT_PROSPECT
ISS Set IO Name	Name	ISS Set IO Name
	Sequence	3
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_NAME
	Set Operator	Set
	Value	ISS_List_Mgmt_Prospective_Contact
ISS Set ID	Name	ISS Set ID
	Sequence	4
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_ID
	Set Operator	Set
	Value	[Id]

Table 58. Actions in ISSSYNC PreDeleteRecord List Mgmt Prospective Contact Action Set

Action	Name of Field	Value
ISS Run WF	Name	ISS Run WF
	Sequence	5
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch PreDelete Record Sync"

ISSSYNC PreWriteRecord List Mgmt Prospective Contact

Table 59 describes the actions in the ISSSYNC PreWriteRecord List Mgmt Prospective Contact action set.

Table 59. Actions in ISSSYNC PreWriteRecord List Mgmt Prospective Contact Action Set

Action	Name of Field	Value
ISS Set System Name	Name	ISS Set System Name
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_SYSTEM
	Set Operator	Set
	Value	SiebelDQ
ISS Set IDT Name	Name	ISS Set IDT Name
	Sequence	2
	Action Type	Attribute Set
	Profile Attribute	IDS_IDT
	Set Operator	Set
	Value	IDS_01_IDT_PROSPECT

Table 59. Actions in ISSSYNC PreWriteRecord List Mgmt Prospective Contact Action Set

Action	Name of Field	Value
ISS Set IO Name	Name	ISS Set IO Name
	Sequence	3
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_NAME
	Set Operator	Set
	Value	ISS_List_Mgmt_Prospective_Contact
ISS Set ID	Name	ISS Set ID
	Sequence	4
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_ID
	Set Operator	Set
	Value	[Id]
ISS Run WF	Name	ISS Run WF
	Sequence	5
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch PreWrite Record Sync"

ISSSYNC WriteRecord List Mgmt Prospective Contact

Table 60 describes the actions in the ISSSYNC WriteRecord List Mgmt Prospective Contact action set.

Table 60. Actions in ISSSYNC WriteRecord List Mgmt Prospective Contact Action Set

Action	Name of Field	Value
ISS Set System Name	Name	ISS Set System Name
	Sequence	1
	Action Type	Attribute Set
	Profile Attribute	IDS_SYSTEM
	Set Operator	Set
	Value	SiebelDQ

Table 60. Actions in ISSSYNC WriteRecord List Mgmt Prospective Contact Action Set

Action	Name of Field	Value
ISS Set IDT Name	Name	ISS Set IDT Name
	Sequence	2
	Action Type	Attribute Set
	Profile Attribute	IDS_IDT
	Set Operator	Set
	Value	IDS_01_IDT_PROSPECT
ISS Set IO Name	Name	ISS Set IO Name
	Sequence	3
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_NAME
	Set Operator	Set
	Value	ISS_List_Mgmt_Prospective_Contact
ISS Set ID	Name	ISS Set ID
	Sequence	4
	Action Type	Attribute Set
	Profile Attribute	IDS_IO_ID
	Set Operator	Set
	Value	[Id]
ISS Set URL	Name	ISS Set URL
	Sequence	5
	Action Type	Attribute Set
	Profile Attribute	IDS_URL
	Set Operator	Set
	Value	"http://SERVERNAME:1671" **NOTE:** Replace SERVERNAME with the Hostname or IP address of computer where XML Sync Server is installed.
ISS Run WF	Name	ISS Run WF
	Sequence	6
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS Launch Write Record Sync"

Siebel Business Applications Generic Action Sets

This topic introduces the following Siebel application generic action sets for all business components:

- "ISSSYNC WriteRecordNew" on page 192
- "ISSSYNC WriteRecordUpdated" on page 192

For more information about creating action sets, including creating actions for action sets, and associating events with action sets, see *Siebel Personalization Administration Guide*.

ISSSYNC WriteRecordNew

Table 61 describes the actions in the ISSSYNC WriteRecordNew action set.

Table 61. Actions in ISSSYNC WriteRecordNew Action Set

Action	Name of Field	Value
ISS Run WF	Name	ISS Run WF
	Sequence	1
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS WriteRecordNew"

ISSSYNC WriteRecordUpdated

Table 62 describes the actions is in the ISSSYNC WriteRecordUpdated action set.

Table 62. Actions in ISSSYNC WriteRecordUpdated Action Set

Action	Name of Field	Value
ISS Run WF	Name	ISS Run WF
	Sequence	1
	Action Type	BusService
	Business Service Name	Workflow Process Manager
	Business Service Method	RunProcess
	Business Service Context	"ProcessName", "ISS WriteRecordUpdated"

F Finding and Using Siebel Data Quality Information

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

- Important Siebel Data Quality Resources on page 193
- SDQ Seed Data on page 195

Important Siebel Data Quality Resources

SDQ is just one of many server processes available in 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 on the Oracle Technology Network" on page 193
- "Third-Party Documentation" on page 194
- "Information about SDQ on My Oracle Support" on page 195

Technical Documentation on the Oracle Technology Network

The following books are on the *Siebel Bookshelf*, available on Oracle Technology Network (OTN). Refer to them when using SDQ:

- *Using Siebel Tools* for information about how to modify standard Siebel CRM objects and create new objects to meet your organization's business requirements.
- *Siebel Fundamentals* for general information about merging records.
- Siebel CRM Deployment Documentation Suite, including:
  - *Siebel Installation Guide* for 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.

  - *Siebel Deployment Planning Guide* to familiarize yourself with the basics of the underlying Siebel application architecture.
  - *Siebel System Monitoring and Diagnostics Guide*.
  - *Going Live with Siebel Business Applications* for information about how to migrate customizations from the development environment to the production environment.
  - *Siebel Security Guide* for information about built-in seed data in the enterprise database, such as employee, position, and organization records.
  - *Siebel Performance Tuning Guide* for information about tuning and monitoring specific areas of the Siebel Business Applications architecture and infrastructure, such as the object manager infrastructure.
- *Siebel Data Model Reference* for information about how data used by Siebel Business 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.
- *Siebel Applications Administration Guide* for general information about administering Siebel Business Applications.
- *Siebel Database Upgrade Guide* or *Siebel Database Upgrade Guide for DB2 for z/OS* for information about upgrading your installation.
- *Siebel System Requirements and Supported Platforms* on Oracle Technology Network for a definitive list of system requirements and supported operating systems 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

Third-Party Documentation

The following third-party documentation, included in *Siebel Business Applications Third-Party Bookshelf* in the product media pack on Oracle E-Delivery, 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 must configure to administer data matching using the SDQ Matching Server.

Information about SDQ on My Oracle Support

The following documentation is on My Oracle Support:

- *Siebel Release Notes* on My Oracle Support. The most current information on known product anomalies and workarounds and any late-breaking information not contained in this book.
- *Maintenance Release Guides*. Important information about updates to applications in maintenance releases. Maintenance release guides are available from My Oracle Support.
- *Technical Notes*. Important information on specific SDQ issues.
- For more important information on various SDQ topics, including time-critical information on key product behaviors and issues, see the following:
  - 476548.1 (Article ID) on Oracle My Oracle Support. This document was previously published as Siebel FAQ 1593.
  - 476974.1 (Article ID) on My Oracle Support. This document was previously published as Siebel FAQ 1843.
  - 476926.1 (Article ID) on My Oracle Support. This document was previously published as Alert 611.

SDQ Seed Data

Oracle's 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 CRM 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 *Siebel Security Guide*.

Index