

# Oracle® Enterprise Data Quality

Customer Data Services Pack Siebel Integration Guide

11g Release 1 (11.1.1.7)

E40741-04

June 2014

---

## 1 Introduction

When integrating a Siebel instance with Oracle Enterprise Data Quality Customer Data Services Pack (EDQ-CDS), Oracle recommends that the necessary components be installed and configured in the following order:

1. Install the EDQ-CDS pack on the EDQ server, see *Oracle Enterprise Data Quality Customer Data Services Pack Installation Guide*.
2. Install the EDQ Siebel Connector on the Siebel server, see *Oracle Enterprise Data Quality Siebel Connector Installation Guide*.
3. Integrate Siebel with EDQ-CDS, as detailed in this guide.

### 1.1 Overview

The EDQ Siebel Connector calls EDQ jobs (for batch matching and health checks) and web services (for real-time cleansing and matching). A shared staging database is used to pass data between Siebel and EDQ-CDS when running batch jobs.

---

---

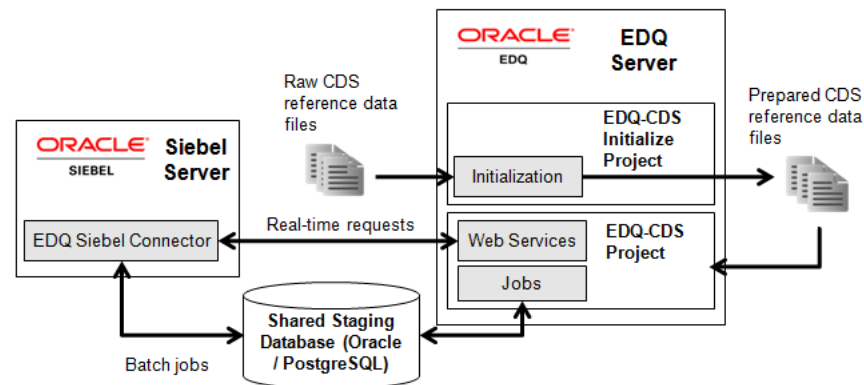
**Note:**

- EDQ-CDS 11g (11.1.1.7) requires the 11g release of the EDQ Siebel Connector.
  - This guide describes the configuration of the Siebel Connector, not the installation. The connector is installed on the Siebel server, and installation instructions vary depending on the operating system of the server. For more information, see *Oracle Enterprise Data Quality Siebel Connector Installation Guide*.
  - Prior to installation, it is important to become familiar with Siebel's universal data quality interface. For more information, see *Siebel Data Quality Administration Guide*
- 
- 

### 1.2 Siebel and EDQ Architecture

The high-level architecture of the EDQ Siebel Connector is illustrated in the following diagram (assuming a single EDQ server):

**Figure 1 EDQ Siebel Connector Architecture**



### 1.3 EDQ with Siebel CRM

EDQ can be deployed to protect and monitor the quality of data in a Siebel Customer Relationship Management (CRM) environment. When attached to Siebel CRM, EDQ can:

- Prevent duplicate contacts, account and prospect records from being added to the system by automatically matching all new and updated records against the other records in the system
- Standardize account, contact, prospect, and address data as it is entered or updated in the system
- Perform batch duplicate identification and cleansing tasks
- Measure the quality of account, contact, prospect, and address data on both an ad-hoc, and a regular basis

As Siebel CRM has only a single set of data to manage, EDQ deployment is simple. When records are added or updated in Siebel CRM, the EDQ real-time interface is used to standardize the data, and then match it against existing data.

The EDQ real-time interface is also used for batch cleansing (standardization) tasks: Siebel sends each record selected for batch processing to the EDQ cleansing web services.

Batch duplicate identification and data quality health check jobs (either Full Batch or Incremental Batch) use the EDQ batch interface. In this case, the driver and candidate records for matching or health checks are written to the shared staging database. EDQ then runs a batch job on these records and when matching occurs the matches are written back to the shared staging database, which Siebel picks up and uses to link records together.

### 1.4 EDQ with Siebel UCM

When deployed with Siebel Universal Customer Master (UCM), EDQ provides the essential matching engine for the hub. As records are added or updated in source systems they are automatically matched against the master data in Siebel UCM in order to maintain the quality of the master data and its linkage with the source data records.

The EDQ services provided to Siebel UCM are the same as those in Siebel CRM though the additional UCM capabilities are used to manage record survivorship and best record generation. Siebel UCM also supports automatic matching when EDQ returns a match score over a specified threshold.

---

---

**Note:** The UCM Batch workflow invokes the EDQ real-time interface for standardization and matching, so that a number of records can be loaded into the hub and automatically matched and reconciled with the existing master data. The EDQ batch interface can still be invoked from Siebel UCM, but works only with the master data records.

---

---

#### 1.4.1 High-Volume Data

When migrating a high volume of data records into Siebel UCM (i.e over 10m records), it is best practice to match and merge the records in EDQ before loading them into Siebel (normally via Siebel's EIM interface), as opposed to matching them using the UCM Batch workflow and the real-time interface. This ensures considerably greater efficiency in the matching process as all in-scope records are matched against each other and all matches found in a single operation, rather than continuously matching each record against the records that have been loaded so far into the hub.

## 2 Configuring the EDQ Siebel Connector

The EDQ Siebel Connector requires a configuration file called `dnd.properties`, which must be installed in the location specified in the `dnd.parms` file (installed and configured as part of the EDQ Siebel Connector.) The settings in the `dnd.properties` file control which EDQ server and project are used to provide data quality services to a Siebel instance, and how to connect to it. To create this file with the settings needed to use EDQ-CDS, use the following procedure:

1. Log in to the Siebel server as the Configuration user (specified during Siebel installation)
2. Change to the directory specified in the `dnd.parms` file.
3. Extract and copy the `dnd.properties` file from the EDQ-CDS installation package into the directory.
4. Use a text editor to open the `dnd.properties` file.
5. Edit the properties as required.

### 2.1 `dnd.properties` Settings

The settings in the `dnd.properties` file fall into the following categories:

- CDS Connection Settings
- Multiple Child Entity groupings
- Real-Time Service Definitions
- Batch Job Definitions

### 2.1.1 CDS Connection Settings

These settings are used to connect to an EDQ server for running jobs and for real-time services. When setting these, make sure there are no trailing spaces in the values as these will cause errors when attempting to establish a connection.

- `httpprefix.cds` - the hostname, HTTP port and context name of the EDQ server (for example `http://hostname:port/edq/webservices`).

---

**Note :** Although EDQ requires SSL to be used for communication between its own web pages and the server, and therefore redirects any http requests to its web pages to the https port, http can still be used for web service requests if required (and allowed by security standards). If EDQ is installed using the Windows installer, the default http port is 9002. On other installations, the port is specified when EDQ is deployed onto the application server. If the use of https is a requirement, the https prefix and port number (which defaults to 9004 if EDQ is installed using the Windows installer), can be specified here, but in order to establish a trusted connection with EDQ, it is also necessary to import either the certificate, or the certificate's root, from the EDQ application server into the Certificate Store of the JRE used by the EDQ Siebel Connector (using the standard Java `keytool` command).

---

- `jmxserver.cds` - the hostname and port of the EDQ server's JMX interface (for example `servername:9005`).

---

**Note:** If EDQ is installed using the Windows installer, the JMX management port defaults to 9005. On other installations, it defaults to 8090. It can be checked by reading the `management.port` entry in the file `director.properties` in the config directory of the EDQ server. If this entry is not present, the port will be 8090.

---

- `username.cds` and `password.cds` - The EDQ login credentials for a designated JMX and Web Services user on the EDQ server. This account must have the following EDQ permissions (at minimum):
  - **System: Connect to Messaging System**, which grants access to submit web service requests or JMS messages to EDQ.
  - **Server Console: Execute Job** and **Director: Execute Job**, to allow the user to execute jobs
  - Permission to the EDQ project stated under `projectname`. To check that the user has project permission, right-click on the project in EDQ's Director application, select **Properties**, then select the **Security** tab. Check that the user is a member of a group with access to the project. If the user account previously described was the account used to import the project from a DXI file, it must have permission.
- **Staging Database connection details** - The connection details and credentials of the Staging Database used to pass data for batch matching jobs between Siebel and EDQ-CDS. For further information, see [Section 4, "Configuring the Staging Database."](#)

- `projectname.cds` - the name of the EDQ-CDS project on the EDQ server. This setting can be left unchanged if the project has not been renamed.
- `projectname.cdshc` - the name of the EDQ-CDS Health Check project on the EDQ server. This setting can be left unchanged if the project has not been renamed.

### 2.1.2 Multiple Child Entity Groupings

Hierarchical data in Siebel is transformed by the EDQ Siebel Connector into a flat record structure so that EDQ can match 'parent' records (such as Contacts, Prospects and Accounts) using the details of multiple 'child' records (such as Names, Addresses, Email Addresses, Alternative Phone Numbers etc.)

The Multiple Child Entity settings specify how scalar (single field) and non-scalar (multi-field) child entities are handled by the Siebel Connector.

---

**Note :** These settings are only required if Multiple Child Entities are enabled in EDQ-CDS and supported by the installed version of Siebel (version 8.1.1.10 or later), see [Section 8, "Vendor Parameters"](#) and the Siebel documentation for further details.

---

#### Scalar Groups

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

The required configuration format is:

```
group.[name] = [BC1],[BC2],...
group.[name].concat = [EDQ-CDS Attribute Name]
group.[name].delimiter = [Delimiter character]
```

where:

- `name`: label applied to the business components.
- `BCn`: Siebel Business Component name.
- `group.[name].concat`: the attribute under which the data will be concatenated.
- `group.[name].delimiter`: the delimiter character used to separate the data.

For example, the following group in the default `dnd.properties` file sets this configuration for the `AlternatePhone` business component in Siebel:

```
group.altphone = AlternatePhone
group.altphone.concat = alternatephone
group.altphone.delimiter = |
```

So all `AlternatePhone` values for each parent entity are concatenated into a delimited list.

This means that `AlternatePhone` values for Contacts in Siebel that are stored in this format:

```
contact1
  AlternatePhone1
  AlternatePhone2
```

```
contact2
  AlternatePhone3
  AlternatePhone4
```

are transformed into this format:

```
contact1 AlternatePhone1|AlternatePhone2
contact2 AlternatePhone3|AlternatePhone4
```

### Non-Scalar Groups

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

The required configuration format is:

```
group. [name] = [BC1], [BC2] ...
```

Where:

- *name*: label applied to the business components
- *BCn*: Siebel Business Component name

For example, the following non-scalar group specifies a list of Siebel Business Components, each storing addresses:

```
group.address = CUTAddress, PersonalAddress
```

This setting means that `CUTAddress` and `PersonalAddress` records are grouped under the address label by the Connector and expanded into separate records for matching in EDQ-CDS.

So, `CUTAddress` and `PersonalAddress` values for Contacts in this format:

```
contact1
  CUTAddress1
  PersonalAddress1
contact2
  CUTAddress2
  PersonalAddress2
```

are transformed into this format:

```
contact1 address1 [CUTAddress1]
contact1 address2 [PersonalAddress1]
contact2 address3 [CUTAddress2]
contact2 address4 [PersonalAddress2]
```

### Real-Time Service Definitions

These properties are used to configure how the Siebel Connector communicates with EDQ-CDS over web services.

---

---

**Note :** It is not normally necessary to change these properties, as the Global Communication settings are inherited.

---

---

The `dnd.properties` file comes with 10 pre-defined Web Services configurations, `ws1` to `ws10`. These are pre-configured with some or all of the following parameters, which can be edited if required:

- `type` - A tag identifying the Data Quality operation in the Siebel vendor parameters.
- `conduit` - Specifies how the Siebel Connector communicates with EDQ-CDS. Possible values are `simplews`, `dbbatch`, `jmsbatch`.
- `url` - The URL end point of the Web Service.
- `failsafe` - Specifies how the Connector behaves in the event of an error:
  - `true`: the Connector fails silently, for example, no error message is generated.
  - `false`: the Connector generates an error message.
- `username` and `password` - The credentials used to connect to the Web Service.
- `parameterNames` - A comma-separated list of Siebel session parameters to be passed to EDQ-CDS as Web Service parameters.
- `idElement` - the name of the XML element generated in the response to Siebel.

### 2.1.3 Batch Job Definitions

The Batch Job Definitions are divided into five groups:

- Account Match
- Account Health Check
- Contact Match
- Contact Health Check
- Prospect Match

These groups all have properties that control the way Batch operations are run.

These properties control how data flows between the Siebel Connector and EDQ-CDS:

- `type` - A tag identifying the Data Quality operation. This must match the in Siebel vendor parameters.
- `conduit` - Specifies how the Siebel Connector communicates with EDQ-CDS. Possible values are `simplews`, `dbbatch`, `jmsbatch`.
- `failsafe` - Specifies how the Connector behaves in the event of an error:
  - `true`: the Connector fails silently, for example, no error message is generated.
  - `false`: the Connector generates an error message.
- `writetable` - the Candidates table in the staging database that the Connector writes data to, and that EDQ-CDS reads data from.
- `db` - The database connection string for the staging database.
- `project` - CDS project name, which reads from the `$projectname` variable at the start of the `dnd.properties` file.
- `job` - The EDQ-CDS job name.
- `runlabel` - The Run Label for the job.
- `overrides` - The values of these settings and the Staging Database override the default (externalized) configuration settings in the EDQ-CDS processes. For example, Cluster Level, Match Threshold (2 and 70 respectively, by default.) However, they can in turn be overridden by parameters configured in Siebel, see [Section 8, "Vendor Parameters."](#)

- readtable - Name of the table in the staging database that matches are written to.
- idelement - The identifier element for the Siebel Business Component.

### 3 Configuring Siebel

The following is a step-by-step guide to configuring Siebel to use EDQ-CDS. Please read the Siebel Data Quality Administration Guide (available as part of Siebel Bookshelf) before attempting to follow these steps.

---

**Note :** Siebel 8.1.1.10 or later is configured by default with all the settings described in this chapter. Therefore, if using this version of Siebel, this chapter can be used as a check list to confirm all the settings are correct. For older versions of Siebel, please use the settings described in this guide except where stated.

---

The guide is in four sections:

- Server Configuration
- Data Quality Administration
- User Preferences
- Creating Templates to Enable Batch Data Quality Jobs (Optional)

#### 3.1 Server Configuration

1. Log into the Siebel web client as a system administrator.
2. Navigate to **Administration - Server Configuration** on the Siebel Site Map.
3. Click **Enterprises** at the top of the page and select the **Component Groups** tab.
4. Find the **Data Quality** component group and ensure it is enabled.
5. Check that the component group has been assigned and enabled on the Siebel server, cai, in the following example:

Component Groups				
<div> <div>Menu</div> <div>New</div> <div>Query</div> <div>Enable</div> <div>Disable</div> </div>				
Component Group	Alias	Number of Components	Enable state	Description
> Data Quality	DataQual	1	Enabled	Data Quality Components
Assignment Management	AsgnMgmt	2	Enabled	Assignment Management Components
Siebel Sales	Sales	2	Enabled	Siebel Sales Components
Siebel eDocuments	eDocuments	1	Enabled	Siebel eDocuments Components
SAP Connector	SAP	5	Enabled	SAP Connector Components
Incentive Compensation	IComp	7	Enabled	Incentive Compensation Components
Siebel Call Center	CallCenter	2	Enabled	Siebel Call Center Components

Components			Component Group Assignments		
<div> <div>Menu</div> <div>Query</div> <div>1 - 1 of 1</div> </div>			<div> <div>Menu</div> <div>Query</div> <div>Assign</div> <div>Unassign</div> <div>Enable</div> </div>		
Component	Alias	Description	Server	Assigned?	Enabled on Server?
> Data Quality Manager	DQMGr	Cleanses data and de	> cai	✓	✓

6. Switch to the **Parameters** tab and set the **Data Cleansing Type** parameter to **EDQ**:



Enterprise Parameters		Menu ▼
Name	Value	
> Data Cleansing Type	EDQ	

- Set the **DeDuplication Data Type** parameter to **EDQ**:

Enterprise Parameters		Menu ▼
Name	Value	
> DeDuplication Data Type	EDQ	

- Click **Servers** at the top of the page, find the **Data Quality Manager** component and select the **Parameters** tab.
- Ensure the following parameter values are set to **EDQ**:
  - Data Cleansing Type
  - DeDuplication Data Type
- Set the following parameter values to **True**:
  - Data Cleansing Enable Flag
  - DeDuplication Enable Flag

Components

Parameters

Events

Menu

Query

Auto Start

Manual Start

Component	Alias	Component Group
> Data Quality Manager	DQMGr	DataQual
File System Manager	FSMSrvr	SystemAux
Server Manager	ServerMgr	System
Server Request Broker	SRBroker	System
Server Request Proxy	SRProc	SystemAux
Server Tables Clean	SvrTblCleanup	SystemAux
Server Task Persistence	SvrTaskPersist	SystemAux

Events

Parameters

Component Parameters

Menu

Query

Reset

Hidden

Parameter	Value	Value on Restart	Default Value
> Data Cleansing Enable	True	True	True
Data Cleansing Type	EDQ	EDQ	EDQ
Data Quality Setting			
DeDuplication Data Type	EDQ	EDQ	EDQ
DeDuplication Enable	True	True	True
Disable DFC		False	

---

**Note :** The preceding illustration shows the values after a server restart, which is performed when Siebel configuration is completed.

---

- Repeat steps 8 to 10 for the Siebel application components you want to use with EDQ-CDS. For example:
  - UCM Object Manager (LANGUAGE)
  - Call Center Object Manager

- EAI Object Manager
- Sales Object Manager

**Tip:** You can use these steps for any other components that you want to use with EDQ-CDS.

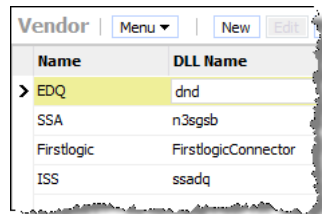
### 3.2 Data Quality Administration

To set the Data Quality Administration options, use the following procedure:

1. Log into Siebel as a system administrator.
2. Navigate to **Administration - Data Quality** on the Siebel Site Map.
3. Click **Data Quality Settings** and create the following options:
  - **Enable DataCleansing = Yes**
  - **Enable DeDuplication = Yes**

Add the following options to enable a pop-up dialog of possible duplicate records for each new record of a given type when the real-time duplicate prevention service is running. If these options are not specified, and the real-time duplicate prevention service is running, the check will still be performed, but you must actively navigate to the **Duplicate Accounts**, **Duplicate Contacts** or **Duplicate Prospects** tab when adding a new record:

- **Force User DeDupe - Account = Yes**
  - **Force User DeDupe - Contact = Yes**
  - **Force User DeDupe - List Mgmt = Yes**
4. Click **Third Party Administration**. Add a new Vendor entry with a Name of **EDQ**, and a DLL Name of **dnd**:



5. Select the EDQ vendor selected and add **BC Operations** as follows:

Business Component Name	Operations
Account	Data Cleansing
Account	DeDuplication
Contact	Data Cleansing
Contact	DeDuplication
CUT Address	Data Cleansing
List Mgmt Prospective Contact	Data Cleansing
List Mgmt Prospective Contact	DeDuplication

Additionally, add the following Business Components to configure matching using multiple child addresses:

Business Component Name	Operations
CUT Address	DeDuplication
Personal Address	Data Cleansing
Personal Address	DeDuplication

6. Scroll to the bottom of the screen and set the **Field Mappings** for each BC Operation. These mappings are detailed in [Section 7, "Field Mappings for Business Components."](#)

---

**Note :** If you add or change field mappings, then they must correspond to the configured Web Services in EDQ (for real-time operations) and the names of the columns in the tables of the staging database (for batch operations). Also, you *must* modify the relevant DQ Integration Objects and Business Services using Siebel Tools. If you are using Siebel UCM, then the you must modify the data maps between UCM Integration Objects and DQ Integration Objects.

---

7. With the EDQ vendor selected, select the **Vendor Parameter** tab in the middle section of the screen and add the parameters specified in [Section 8, "Vendor Parameters."](#)

### 3.3 User Preferences

To enable data quality for any user, use the following procedure:

1. Select **User Preferences** on the **Tools** menu.
2. Click **Data Quality**.
3. Set both the Enable DeDuplication and Enable Data Cleansing options to **Yes**.

This completes the Siebel configuration. Restart the Siebel server to ensure that all the configuration changes have taken effect.

### 3.4 (Optional) Creating Templates to Enable Batch Data Quality Jobs

Siebel can be configured to run batch data quality jobs from the Server Management UI. Jobs can also be run from the Siebel command line, or job configurations can be stored in files and reused as required.

To do this, some custom Job Templates must be added to Siebel. These templates and the parameters required are listed in [Section 6, "Job Template Configuration."](#)

To add a Job Template:

1. Open the Siebel web client.
2. Navigate to **Administration - Server Configuration** on the Siebel Site Map.
3. Click **Job Templates**.

4. Click **New** to create a new Job Template.
5. Complete the fields in the Job Templates and Job Parameters area using the details provided in [Section 6, "Job Template Configuration."](#)

---

**Note :** Set the Component field to Data Quality Manager for each new Job Template. If there are no options listed in this drop-down field, navigate to the **Component Definitions** tab on the **Enterprise** screen, and click Synchronize.

---

Create a new Job Template for every job listed in [Section 6, "Job Template Configuration."](#)

## 4 Configuring the Staging Database

The Staging Database is used by the Siebel Connector as a staging area for handing over data between Siebel and EDQ when running batch jobs through Siebel's Data Quality Manager.

This batch interface is most commonly used when EDQ is connected to a standalone Siebel CRM system. In Siebel UCM, the 'UCM Batch' flow does not use this interface when matching inbound data against the master data in Siebel UCM. Rather, it calls out to EDQ's real-time matching services.

However, for both CRM and UCM the Siebel Data Quality Manager service can run batch duplicate identification and health check jobs on the master data only.

When such batch jobs are run, driver and candidate records for matching and input records for the Data Quality Health Check service are written to tables in the staging database, to be read by an EDQ job. For the matching service only, EDQ then exports duplicates from the matching process to another table in this staging database to be read back into Siebel.

### 4.1 Creating Tables

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

SQL Script	Table Created
<code>siebel_account_candidates.sql</code>	<code>SIEBEL_ACCOUNT_CANDIDATES</code>
<code>siebel_account_matches.sql</code>	<code>SIEBEL_ACCOUNT_MATCHES</code>
<code>siebel_contact_candidates.sql</code>	<code>SIEBEL_CONTACT_CANDIDATES</code>
<code>siebel_contact_matches.sql</code>	<code>SIEBEL_CONTACT_MATCHES</code>
<code>siebel_prospect_candidates.sql</code>	<code>SIEBEL_PROSPECT_CANDIDATES</code>
<code>siebel_prospect_matches.sql</code>	<code>SIEBEL_PROSPECT_MATCHES</code>
<code>siebel_account_health_check.sql</code>	<code>SIEBEL_ACCOUNT_HEALTH_CHECK</code>
<code>siebel_contact_health_check.sql</code>	<code>SIEBEL_CONTACT_HEALTH_CHECK</code>

## 4.2 Configuring Connections

Both the Siebel Connector and EDQ-CDS itself need to connect to the Staging Database in order to read and write to the Candidate and Match tables when processing Batch jobs. These tables can be created in any schema of a supported database type (Oracle or PostgreSQL). The default connection string is for an Oracle database.

The connection details are specified in the `dnd.properties` file, see [Section 2, "Configuring the EDQ Siebel Connector."](#) To configure these, open the `dnd.properties` file and edit the parameters in the **CDS Connection Settings** section near the top of the file. These parameters control the database host, port, credentials and other settings used to connect to the Staging Database.

## 5 Finalizing and Verifying the Installation

The following sections describe how to finalize the system and verify that the EDQ Siebel Connector is installed and configured correctly.

Before starting, ensure that the appropriate real-time jobs are running in EDQ, see *Oracle Enterprise Data Quality Customer Data Services Installation Guide*.

### 5.1 Key Generation

If the Siebel database is already populated with records (Accounts/Contacts/Prospects) then the cluster keys in the system must be generated or refreshed for the new configuration before it is used.

In Siebel, run the following jobs (as defined in [Section 6, "Job Template Configuration"](#)) from the **Administration > Server** page:

- Generate account keys
- Generate contact keys
- Generate prospect keys

---

**Note :** If EDQ is being used for key generation, these jobs will automatically call EDQ clustering web services instead of using the configured Token Expression in Siebel to generate the keys.

---

### 5.2 Testing Account Cleaning

To test the cleaning service for Account records:

1. Check that data cleansing is enabled for the server by ensuring that **Administration - Data Quality > Data Quality Settings > Enable Data Cleansing** is set to "Yes".

---

**Note :** If this setting is changed from No to Yes, it is necessary to restart the Siebel Server.

---

2. Check that data cleansing is enabled for the current user profile by ensuring that **Tools > User Preferences > Data Quality > Enable Data Cleansing** is set to "Yes".

---

**Note :** If this setting is changed from No to Yes, it is necessary to exit the current client session and log in again as the same user.

---

3. Enter a new Account record with an Account Name in lower case letters.

Account Name:*	datanomic ltd	Site:	
Address:		State:	
City:		Country:	
Zip Code:			

4. Save the record.
5. Check that the Account Name has been converted to upper case to verify that the cleaning service has been called.

Account Name:*	DATANOMIC LTD	Site:	
Address:		State:	
City:		Country:	
Zip Code:			

### 5.3 Testing Batch Contact Matching

To test the batch matching service for Contact records:

1. Temporarily disable real-time deduplication for the current user profile by setting **Tools > User Preferences > Data Quality > Enable DeDuplication** to **No**.

---

**Note :** If this setting is changed from No to Yes it is necessary to restart the Siebel Server.

---

2. Enter and save two new Contact records with the same name and email address.

SMITH	Peter	p.smith@mymail.com
> SMITH	Peter	p.smith@mymail.com

**Peter SMITH**

Menu ▾	New	Delete	Query
--------	-----	--------	-------

Last Name:*	SMITH	Account:	
First Name:*	Peter	Job Title:	
Middle:		Work #:	
Mr/Ms:		Home #:	
Suffix:		Email:	p.smith@mymail.com

3. Run the Batch contact match job from the **Administration - Server** page
4. Verify that the records were matched by checking the **Administration - Data Quality > Duplicate Contacts** page.

Rules | Duplicate Accounts | Duplicate Contacts

Duplicate Contacts | Menu | Query

Last Name	First Name	MI	Job Title	Account	Site	Comment
> SMITH	Peter					
SMITH	Peter					

## 5.4 Testing Real-Time Contact Matching

To test the real-time matching service for Contact records:

1. Check that real-time deduplication is enabled for the current user profile by checking that **Tools > User Preferences > Data Quality > Enable DeDuplication** is set to **Yes**.
2. Set **Administration - Data Quality > Data Quality Settings > Force User DeDupe - Contact** to **Yes**.
3. Enter a new Contact record with the same details as in [Section 5.3, "Testing Batch Contact Matching"](#).
4. Save the record.
5. Verify that the **Possible Matching Contacts** dialog is displayed showing matches against the two contact records created previously.

Possible Matching Contacts - Windows Internet Explorer

Pick

1 - 2 of 2

Score ▾	Account	Site	Last Name	First Name	MI
> 96			SMITH	Peter	
96			SMITH	Peter	

---

**Note:** An exact name and email match, as in this example, will only match if EDQ is being used for key generation (for Siebel 8.1.1.10 or later), or if the Query and Token expressions used for key generation have been adjusted to use only name, or name and email, attributes.

---

## 6 Job Template Configuration

---

**Note :** For clarity, the Data Quality Setting parameter found in most of the Templates listed in the table begins with a single double quote mark (") and two single (') quote marks, and ends with one single quote mark and one double.

---

Name	Short Name	Parameters
Batch account cleanse	BatAccClean	Buscomp Name = Account Business Object Name = Account Operation Type = Data Cleansing

<b>Name</b>	<b>Short Name</b>	<b>Parameters</b>
Batch account match	BatAccMatch	Buscomp Name = Account Business Object Name = Account Data Quality Setting = ''','Yes','dedupe''' Operation Type = DeDuplication
Batch account health check	BatAccHealth	Buscomp Name = Account Business Object Name = Account Data Quality Setting = ''','Yes','healthcheck''' Operation Type = DeDuplication
Batch address cleanse	BatAddClean	Buscomp Name = CUT Address Business Object Name = CUT Address Operation Type = Data Cleansing
Batch contact cleanse	BatConClean	Buscomp Name = Contact Business Object Name = Contact Operation Type = Data Cleansing
Batch contact match	BatConMatch	Buscomp Name = Contact Business Object Name = Contact Data Quality Setting = ''','Yes','dedupe''' Operation Type = DeDuplication
Batch contact health check	BatConHealth	Buscomp Name = Contact Business Object Name = Contact Data Quality Setting = ''','Yes','healthcheck''' Operation Type = DeDuplication
Batch prospect cleanse	BatProClean	Buscomp Name = List Mgmt Prospective Contact Business Object Name = List Mgmt Prospective Contact Operation Type = Data Cleansing
Batch prospect match	BatProMatch	Buscomp Name = List Mgmt Prospective Contact Business Object Name =List Mgmt Prospective Contact Data Quality Setting = ''','Yes','dedupe''' Operation Type = DeDuplication
Generate account keys	GenAccKey	Buscomp Name = Account Business Object Name = Account Operation Type = Key Generate
Generate contact keys	GenConKey	Buscomp Name = Contact Business Object Name = Contact Operation Type = Key Generate



Name	Short Name	Parameters
Generate prospect keys	GenProKey	Buscomp Name = List Mgmt Prospective Contact Business Object Name = List Mgmt Prospective Contact Operation Type = Key Generate
Incremental account match	IncAccMatch	Buscomp Name = Account B Business Object Name = Account Data Quality Setting = ''', 'No', 'dedupe''' Operation Type = DeDuplication Object Where Clause = [Updated] > '12/18/2007 00:00:00' <b>Note:</b> The preceding Object Where Clause is an example only.
Incremental account health check	IncAccHealth	Buscomp Name = Account Business Object Name = Account Data Quality Setting = ''', 'No', 'healthcheck''' Operation Type = DeDuplication Object Where Clause = [Updated] > '12/18/2007 00:00:00'
Incremental contact match	IncConMatch	Buscomp Name = Contact Business Object Name = Contact Data Quality Setting = ''', 'No', 'dedupe''' Operation Type = DeDuplication Object Where Clause = [Updated] > '12/18/2007 00:00:00' <b>Note:</b> The preceding Object Where Clause is an example only
Incremental contact health check	IncConHealth	Buscomp Name = Contact Business Object Name = Contact Data Quality Setting = ''', 'No', 'healthcheck''' Operation Type = DeDuplication Object Where Clause = [Updated] > '12/18/2007 00:00:00'
Incremental prospect match	IncProMatch	Buscomp Name = List Mgmt Prospective Contact Business Object Name = List Mgmt Prospective Contact Data Quality Setting = ''', 'No', 'dedupe''' Operation Type = DeDuplication Object Where Clause = [Updated] > '12/18/2007 00:00:00' <b>Note:</b> The preceding Object Where Clause is an example only.

Name	Short Name	Parameters
Refresh account keys	RefAccKey	Buscomp Name = Account Business Object Name = Account Operation Type = Key Refresh
Refresh contact keys	RefConKey	Buscomp Name = Contact Business Object Name = Contact Operation Type = Key Refresh
Refresh prospect keys	RefProKey	Buscomp Name = List Mgmt Prospective Contact Business Object Name = List Mgmt Prospective Contact Operation Type = Key Refresh

## 7 Field Mappings for Business Components

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

---

**Note :** All fields in these Business Components are drop-down fields, with the exception of Id. This field must be completed manually.

---

### 7.1 Account - Data Cleansing

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

### 7.2 Account - DeDuplication

Business Component Field	Mapped Field
Id	entityid
Language Code	languages
Name	name

Business Component Field	Mapped Field
Location	subname
Main Phone Number	phone
Tax ID Number	taxnumber
VAT registration number	vatnumber
Home Page	website

The following fields are only used if Multiple Child Entities are disabled:

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

### 7.3 CUT Address - Data Cleansing

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

### 7.4 CUT Address - DeDuplication

---

**Note :** This Business Component is only used if Multiple Child Entities are enabled.

---

Business Component Field	Mapped Field
Street Address	address1
Street Address 2	address2
City	city

Business Component Field	Mapped Field
State	adminarea
Postal Code	postalcode
Country	country

## 7.5 Personal Address - Data Cleansing

---

**Note :** This Business Component is only used if Multiple Child Entities are enabled.

---

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

## 7.6 Personal Address - DeDuplication

---

**Note :** This Business Component is only used if Multiple Child Entities are enabled.

---

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

## 7.7 Contact - Data Cleansing

Business Component Field	Mapped Field
Id	individualid
Language Code	languages
M/M	title
M/F	gender

<b>Business Component Field</b>	<b>Mapped Field</b>
First Name	firstname
Middle Name	middlename
Last Name	lastname
Home Phone #	homephone
Work Phone #	workphone
Fax Phone #	faxphone
Cellular Phone #	mobilephone
Email Address	email
Job Title	jobtitle
Social Security Number	taxnumber
Birth Date	dob

## 7.8 Contact - DeDuplication

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

The following fields are only used if Multiple Child Entities are disabled:

<b>Business Component Field</b>	<b>Mapped Field</b>
Primary Street Address	address1
Primary Address Street Address2	address2

Business Component Field	Mapped Field
Primary Address Street Address3	address3
Primary City	city
Primary State	adminarea
Primary Postal Code	postalcode
Primary Country	country

## 7.9 List Mgmt Prospective Contact - Data Cleansing

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

## 7.10 List Mgmt Prospective Contact - DeDuplication

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

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

The following fields are only used if Multiple Child Entities are disabled:

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

## 8 Vendor Parameters

Name	Value
Key Generation Process by Third Party	<p>Yes, if using EDQ web services for key generation. No, if using Siebel Token expressions.</p> <p><b>Note:</b> This parameter should be set to "Yes" if external key generation is supported by the installed version of Siebel (version 8.1.1.10 or later). If the parameter is set to Yes, the Query and Token Expression parameters will be ignored, as these will not control key generation or candidate selection for real-time matching. For earlier versions of Siebel this parameter must be set to No and the Query and Token Expression Parameters must be set.</p>
Support Multiple Child Entities Deduplication	<p>Yes, if matching using multiple child entities. Otherwise, No.</p> <p><b>Note:</b> If this parameter is set to Yes, the CUT and Personal Address DeDup Record Type parameters are enabled. This parameter should only be set to Yes if Multiple Child Entities are supported by the installed version of Siebel (version 8.1.1.10 or later).</p>
Account DataCleanse Record Type	accountclean
Account DeDup Record Type	accountmatch
CUT Address DataCleanse Record Type	addressclean
CUT Address DeDup Record Type	<p>addressmatch</p> <p><b>Note:</b> Only used if multiple child entities are enabled.</p>

Name	Value
Personal Address DataCleanse Record Type	addressclean
Personal Address DeDup Record Type	addressmatch <b>Note:</b> Only used if multiple child entities are enabled.
Contact DataCleanse Record Type	contactclean
Contact DeDup Record Type	contactmatch
List Mgmt Prospective Contact DataCleanse Record Type	prospectclean
List Mgmt Prospective Contact DeDup Record Type	prospectmatch
Batch Max Num Of Records	200
Parameter 1	"session", "clusterlevel", "2" <b>Note:</b> There must be a space after each comma.
Parameter 2	"session", "matchthreshold", "70" <b>Note:</b> There must be a space after each comma.

## 8.1 Query and Token Expression Parameters

As mentioned in [Section 8, "Vendor Parameters"](#), these parameters should only be configured if keys are generated by Siebel, rather than by calling EDQ key generation services. This applies to versions of Siebel prior to 8.1.1.10.

If the installed version of Siebel supports the use of EDQ services for key generation, then these services should be used to ensure the right balance between performance and match effectiveness.

---

**Note :** The values of these Query and Token parameters are provided as examples only. They assume that there is a good level of completion of address data for individuals, that postal code on the primary address, and that most individuals are assigned to accounts. If this is not the case for the available data, these parameters must be configured to use the most frequently populated identifying data fields. For more information, see *Oracle Siebel Data Quality Administration Guide*.

---

Name	Value
Account Query Expression	"IfNull (Left ([Primary Account Postal Code], 5), '?????') + IfNull (Left ([Name], 1), '?') + IfNull (Mid([Street Address], FindNoneOf ([Street Address], '1234567890 '), 1), '?')"
Account Token Expression	"IfNull (Left ([Primary Account Postal Code], 5), '_____') + IfNull (Left ([Name], 1), '_') + IfNull (Mid([Street Address], FindNoneOf ([Street Address], '1234567890 '), 1), '_')"



Name	Value
Contact Query Expression	"IfNull (Left ([Postal Code], 5), '?????') + IfNull(Left ([Account], 1), '?') + IfNull (Left ([Last Name], 1), '?')"
Contact Token Expression	"IfNull (Left ([Postal Code], 5), '_____') + IfNull(Left ([Account], 1), '_') + IfNull (Left ([Last Name], 1), '_')"
List Mgmt Prospective Contact Query Expression	"IfNull (Left ([Postal Code], 5), '?????') + IfNull(Left ([Account], 1), '?') + IfNull (Left ([Last Name], 1), '?')"
List Mgmt Prospective Contact Token Expression	"IfNull (Left ([Postal Code], 5), '_____') + IfNull(Left ([Account], 1), '_') + IfNull (Left ([Last Name], 1), '_')"

---

**Note :** The preceding values in the query, token expressions and parameter names are case sensitive. Also, '\_\_\_\_\_' in the Account Token Expression and Contact Token Expression settings represents five underscore characters in a sequence.

---

## 9 Related Documents

For more information, see the following documents in the Oracle Enterprise Data Quality documentation set:

- *Oracle Enterprise Data Quality Release Notes*
- *Oracle Enterprise Data Quality Installation Guide*
- *Oracle Enterprise Data Quality Architecture Guide*
- *Oracle Enterprise Data Quality Siebel Connector Installation Guide*
- *Oracle Enterprise Data Quality Customer Data Services Pack Installation Guide*
- *Oracle Enterprise Data Quality Customer Data Services Pack Siebel Integration Guide*
- *Oracle Enterprise Data Quality Customer Data Services Pack Matching Guide*
- *Oracle Enterprise Data Quality Customer Data Services Pack Data Quality Health Check Guide*
- *Oracle Enterprise Data Quality Customer Data Services Pack Customization Guide*
- *Oracle Enterprise Data Quality Customer Data Services Pack Business Services Guide*

See the latest version of this and all documents in the Oracle Enterprise Data Quality Documentation website at

[http://download.oracle.com/docs/cd/E48549\\_01/index.htm](http://download.oracle.com/docs/cd/E48549_01/index.htm)

## 10 Documentation Accessibility

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

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

## Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

---

Customer Data Services Pack Siebel Integration Guide, 11g Release 1 (11.1.1.7)  
E40741-04

Copyright © 2006, 2014, Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.