

Oracle® Enterprise Data Quality

Customer Data Services Pack Business Services Guide

Release 11g R1 (11.1.1.7)

E40733-02

October 2013

1 Introduction

Oracle Enterprise Data Quality Customer Data Services Pack (EDQ-CDS) can be used as a standalone addition to EDQ, or it can be called by third-party applications.

The provided business services are ready for integration with Siebel Customer Relationship Management (CRM) or Universal Customer Master (UCM) but may also be called by other applications if they are configured to do so.

Ready-to-use, EDQ-CDS provides pre-configured data quality services which can be modified or enhanced by editing the underlying EDQ processes that implement their functionality. These three types of service can be used with Individual, Entity, and Address records:

- Cleaning
- Clustering (for use in Matching integrations)
- Matching

Note: It is possible to further extend the functionality of the EDQ-CDS pack by installing the Data Quality Health Check project, which allows users to perform batch data quality checking of customer data. For more information, see *Oracle Enterprise Data Quality Customer Data Services Pack Health Check Guide*.

2 Cleaning Services

There are three cleaning services provided in EDQ-CDS: Address Clean, Individual Clean and Entity Clean. The Individual and Entity Clean services are provided as placeholders, which are pre-integrated with Siebel, and easily integrated with other applications, but which need to be modified towards specific requirements.

2.1 Address Clean

The EDQ-CDS Address Clean web service provides the following functionality:

- Verification of an input address (returning a verification code and description)
- Geocoding of an address (returning latitude and longitude co-ordinates, with additional metadata)

- Correction, standardization and completion of input addresses (provided the address was verified to a sufficient, configurable, level)

The service is N:N; that is, single and multiple record input and output is possible, but only one record is returned for each record submitted. Each input address is verified and may be corrected, enhanced and geocoded, depending on the options that the job is run with, and the input parameters.

Siebel's Data Quality interface always calls the service with a single record per request, whether running in real-time or batch.

2.1.1 Use

The Address Clean web service is normally used for real-time verification and cleansing of addresses as they are entered and updated in an application, such as Siebel.

In the case of Siebel, the web service is also used in batch. When a batch address cleansing job is run, the web service will be used on all of the in-scope records in the batch job.

For other applications, it is recommended to add configuration to EDQ-CDS to map data from and to the Address Clean data interface in order to run the service in batch mode.

2.1.2 Interface

The following table provides a guide to the interface attributes of the Address Clean web service.

| Attribute Name | Data Type | Use | Notes |
|-------------------------|-----------|--------|---|
| addressid | String | In/Out | Unique identifier for the address. |
| address1 | String | In/Out | Address line 1 |
| address2 | String | In/Out | Address line 2 |
| address3 | String | In/Out | Address line 3 |
| address4 | String | In/Out | Address line 4 |
| dependentlocality | String | In/Out | A smaller population center data element, dependent on the contents of the <code>city</code> field. For example, a Neighborhood in Turkey. For many countries, this attribute is not used. |
| doubledependentlocality | String | In/Out | The smallest population center data element, dependent on both the contents of the <code>city</code> and <code>dependentlocality</code> fields. For example, a village in the UK. For many countries, this attribute is not used. |
| city | String | In/Out | The locality, town or city of the address. |
| subadminarea | String | In/Out | The smallest geographic data element within a country. For example, a county in the USA. |

| Attribute Name | Data Type | Use | Notes |
|---|-----------|------------|---|
| adminarea | String | In/Out | The most common geographic data element within a country. For example, a State in the USA or a Province in Canada. |
| postalcode | String | In/Out | Postal or zip code for the address, if relevant for the country. |
| country | String | In/Out | On input, an ISO two-character country code (preferred) or country name. On output, the full country name (even if the input is the country ISO code). Note: If the country field is blank, the service attempts to derive the country from the city attribute. When this is not present, the default specified in the Run Profile is used. |
| defaultcountrycode | String | Input only | Default ISO two-character country code to use if country not populated. This overrides the default value used when running the job. |
| case (parameter) | String | Input only | Transforms output according to the setting: U (Upper)- Transforms all text to Upper case. L (Lower)- Transforms all text to Lower case. M (Mixed) - Transform all text to Mixed case, except postalcode and adminarea. These field values are left as returned from the AV processor if the address was verified. If the address was not verified, the postalcode is left as entered, and the adminarea is converted to Mixed case. O (Original) - Text is not transformed. |
| mode (parameter) | String | Input only | Mode in which the request is to be run. Currently, only V (Verify) is supported. |
| minimumverificationmatchscore (parameter) | Number | Input only | A numeric value between 0 and 100, representing the minimum score which a match must achieve to be used as a cleaned address. Input addresses will be left unchanged if the Match Score of the Address Verification processor for the address is lower than the input value. |

| Attribute Name | Data Type | Use | Notes |
|---|-----------|-------------|--|
| minimumverificationlevel (parameter) | Number | Input only | <p>A numeric value between 1 and 5, representing the minimum verification level which a match must achieve to be used as a cleaned address.</p> <p>Input addresses will be left unchanged if the Verification Level of the Address Verification processor for the address is lower than the input value. For a description of what each level means, see Section , "Notes on Verification Levels".</p> |
| allowedverificationresultcodes (parameter) | String | Input only | <p>A list of any of the following single-letter result codes with no separator (for example, 'VPA'):</p> <p>V (Verified), P (Partially Verified) A (Ambiguous) R (Reverted) U (Unverified)</p> <p>Input addresses will be left unchanged if the Verification Result Code. For example, the first character of the verificationcode) for the address is not one of the listed input values.</p> |
| fulladdress | String | Output only | Full verified address returned from address verification. The address lines are pipe-separated. |
| countrycode | String | Output only | ISO 2 char country code of verified address country. |
| verificationcode | String | Output only | Verification code for the address. See the online help for the EDQ Address Verification processor for more information on how to understand the returned code. |
| verified | String | Output only | Whether the result produced was verified to a sufficient level, according to the configuration of the process and the resultant verification code. This also indicates whether or not the input address was changed to the returned address from address verification. Possible returned values are Y (verified), N (not verified) or X (EDQ Address Verification is not installed). |
| verificationcodedescription | String | Output only | US English description of the verification code. |
| latitude | Number | Output only | WGS 84 latitude in decimal degrees format. |

| Attribute Name | Data Type | Use | Notes |
|----------------------------|-----------|-------------|---|
| longitude | Number | Output only | WGS 84 longitude in decimal degrees format. |
| geoaccuracycode | String | Output only | A code indicating the level of accuracy of the returned geocodes (latitude and longitude) co-ordinates. See the online help for the EDQ Address Verification processor for more information on how to understand the returned code. |
| geoaccuracycodedescription | String | Output only | US English description of the geoaccuracycode. |
| geodistance | Number | Output only | Radius of accuracy (in meters) for the returned geocodes. The higher the value, the less accurate the geocoding result. |

2.1.3 Parameters

The Address Clean web service uses a number of input parameters to control its behavior when processing addresses, as listed and described in the table above.

The `minimumverificationmatchscore`, `minimumverificationlevel` and `allowedverificationresultcodes` parameters are all used as message-level thresholds to override whether or not to change (clean) an input address based on the confidence level that the EDQ Address Verification processor reaches when processing it. Normally, and when using the Address Clean service with Siebel, these parameters are not used, and the underlying settings in the Address Clean process are used to drive whether or not to change the address. In this process it is possible to set these same parameters on a per-country basis if required. Where country-specific thresholds are not provided, global default settings are applied, and these may be set using the EDQ-CDS Run Profile. The priority in which the thresholds are applied is therefore:

1. Per-message threshold settings using the parameter attributes as above
2. Per-country threshold values expressed in the Reference Data set `Address Clean - Country verification level and results`
3. The global default settings expressed in the process and overridden on a per-run basis by the use of a run profile.

An additional configuration option is available to control the number of address lines that are returned from the service. This is not exposed as a parameter on the interface, but can be set using the phase `*.process.Clean\ -\ Address.Number\ Of\ Address\ Lines` Run Profile setting. The default number of lines to return is 4.

Notes on Verification Levels

The following verification levels are possible. The maximum verification level that it is possible to reach varies by country. For information on the maximum level in each country, see the Loqate Oracle EDQ Portal website at

<http://www.loqate.com/oracle>

The verification level is output as the second character of the Accuracy Code returned by the EDQ Address Verification processor. The 'post-processed' verification level is

used (not the 'pre-processed' level); that is, the verification level achieved after EDQ Address Verification applies standardization and parsing to the input address.

| Verification Level | Description |
|--------------------|---|
| 1 | Verified to Administrative Area (State, Region or County) level |
| 2 | Verified to Locality (City or Town) level |
| 3 | Verified to Thoroughfare (Street) level |
| 4 | Verified to Premise (Building Number) level |
| 5 | Verified to Delivery Point (Sub-Building Number) level |

Note: If EDQ Address Verification is not installed (or not installed correctly), the Address Clean service can still be installed, and the job that implements it can still be run. However, if a request is made to the service, all the output fields will be blank, except for the `verified` output field, which will have the value `X`, and the `verificationcode` output, which will have the value `-1.0`.

2.2 Individual Clean

The Individual Clean web service is designed to verify, correct, standardize or enhance records representing individuals, whether these be customers, prospective customers, contacts, or employees.

The **Clean - Individual** process that implements this service in EDQ-CDS is just a placeholder, and must be customized to requirements. A default process that converts the input name attributes to upper case is provided so that when connecting this service to Siebel or other applications, it is simple to test that the service is correctly connected.

The service is N:N; that is, single and multiple record input and output is possible, but only one record is returned for each record submitted.

The Siebel Data Quality interface always calls the service with a single record per request, whether running in real-time or batch.

2.2.1 Use

The Individual Clean web service may be extended for many purposes, including (but not limited to):

- Verification of input details related to individuals (for example, an email address)
- Standardization of input details related to individuals (for example, a job title)
- Enhancement of data related to individuals (for example, by matching reference data for individuals and returning additional attributes, such as social media handles)

Normally, the web service will be called in real-time, when individual records are added or updated in an application.

In the case of Siebel, the web service is also used in batch. When a batch contact cleansing job is run, the web service will be used on all of the in-scope records in the batch job.

For other applications, it is recommended to add configuration to EDQ-CDS to map data from and to the **Individual Clean** data interface in order to run the service in batch mode.

The interface used by the service is designed to map directly to the Contact business component in Siebel, but can be freely extended with new attributes on both input and output. Siebel's DQ vendor parameters may be extended to pass through different attributes to the service.

2.2.2 Interface

The following table provides a guide to the default Individual Clean web service interface. All attributes are both input and output by default. It is possible to input an empty value to the interface but to populate the attribute on output, so providing a data enhancement service.

| Attribute Name | Data Type | Notes |
|------------------|-----------|---|
| individualid | String | A unique identifier for the individual record. |
| languages | String | 3 character Siebel language code. This can be used to determine whether a name containing Kanji should be treated as Japanese or Chinese. |
| nameid | String | Unique identifier for the name. |
| title | String | Title |
| firstname | String | First name |
| middlename | String | Middle name |
| lastname | String | Last Name |
| gender | String | M or F |
| dob | String | Date of Birth |
| jobtitle | String | Job Title |
| homephone | String | Home Phone Number |
| workphone | String | Work Phone Number |
| mobilephone | String | Mobile Phone Number |
| faxphone | String | Fax Number |
| alternatephone | String | Alternate Phone Number |
| email | String | Email Address |
| taxnumber | String | Tax Number |
| nationalidnumber | String | Social Security Number (US) or equivalent. |

2.3 Entity Clean

The Entity Clean web service is designed to verify, correct, standardize or enhance records representing entities, whether these be company customers, prospective company customers, suppliers, or other organizations.

The **Clean - Entity** process that implements this service in EDQ-CDS is just a placeholder, and must be customized to requirements. A default process that converts the input name and surname attributes to upper case is provided so that when

connecting this service to Siebel or other applications, it is simple to test that the service is correctly connected.

The service is N:N; that is, single and multiple record input and output is possible, but only one record is returned for each record submitted.

The Siebel Data Quality interface always calls the service with a single record per request, whether running in real-time or batch.

2.3.1 Use

The Entity Clean web service may be extended for many purposes, including (but not limited to):

- Verification of input details related to entities (for example to check that a website is syntactically valid)
- Standardization of input details related to entities (for example company names and locations)
- Enhancement of data related to entities (for example by matching reference data for entities and returning additional attributes, such as DUNS numbers from Dun and Bradstreet)

Normally, the web service will be called in real-time, when entity records are added or updated in an application.

In the case of Siebel, the web service is also used in batch. When a batch account cleansing job is run, the web service will be used on all of the in-scope records in the batch job.

For other applications, it is recommended to add configuration to EDQ-CDS to map data from and to the **Entity Clean** data interface in order to run the service in batch mode.

The interface used by the service is designed to map directly to the Account business component in Siebel, but can be freely extended with new attributes on both input and output. Siebel's Data Quality vendor parameters may be extended to pass through different attributes to the service.

2.3.2 Interface

The following table provides a guide to the default Entity Clean web service interface. All attributes are both input and output by default. It is possible to input an empty value to the interface but to populate the attribute on output, so providing a data enhancement service.

| Attribute Name | Data Type | Notes |
|----------------|-----------|---|
| entityid | String | A unique identifier for the entity record. |
| languages | String | 3 character Siebel language code. This can be used to determine whether a name containing Kanji should be treated as Japanese or Chinese. |
| nameid | String | Unique identifier for the name. |
| name | String | Organization name for example, "Oracle Corporation UK". |
| subname | String | Department or site for example, "Reading", "Accounts Payable", etc. |

| Attribute Name | Data Type | Notes |
|----------------|-----------|------------------------|
| phone | String | Phone Number |
| alternatephone | String | Alternate Phone Number |
| website | String | Website Address |
| taxnumber | String | Company Tax Number |
| vatnumber | String | Company VAT Number |

3 Clustering Services

EDQ-CDS clustering services are designed to generate a number of key values for an individual, entity or address record. The returned key values for a record are then used by applications such as Siebel to select 'candidate' records for matching, where any existing record that shares any key value with the 'driving' record (the record submitted to the clustering service) should be considered a candidate. The driving and candidate records are then submitted in a single request to the relevant Matching service.

In addition to being called in real-time in order to prevent the insertion of duplicate records into an application, the clustering services are used in batch mode to populate the key values on all existing records in the application, so that real-time and incremental batch matching jobs, both of which use the key values for existing records for candidate selection, work correctly.

The clustering services are N:N; meaning that single and multiple record input and output is possible. In real-time, a single output record is returned containing an array of keys. In batch mode, each key is returned as a separate record in the staging table.

3.1 Use

The real-time clustering services are normally used as the first call to EDQ-CDS when a new or updated record needs to be checked for matches against records in an application. The returned key values are used to select candidate records to be submitted with the driving record to the matching service.

In order to ensure that keys are always up-to-date, the clustering services should be called whenever a record is updated. This includes the scenario when a master record in Siebel UCM, or other hub, is updated due to a confirmed match with an incoming driver record.

The clustering services are exposed using the following:

Web Services:

- IndividualCluster
- EntityCluster
- AddressCluster

Batch Jobs:

- Batch Individual Cluster
- Batch Entity Cluster
- Batch Address Cluster

3.2 Interface

The clustering web services present input interfaces with direct mappings to the shared 'Candidate' data interfaces as follows:

| Web Service | Input Interface |
|-------------------|---|
| IndividualCluster | See Section 5.1.1, "Individual Candidates." |
| EntityCluster | See Section 5.1.2, "Entity Candidates." |
| AddressCluster | See Section 5.1.3, "Address Candidates." |

All the clustering web services return output attributes using the common Real-time Cluster Results Interface data interface, see [Section 5.3, "Cluster Results Interfaces."](#)

3.3 Parameters

The IndividualCluster, EntityCluster and AddressCluster web services all expose a clusterlevel parameter attribute, which is used to drive the sensitivity of the clustering service:

| Parameter Attributes | Data Type | Accepted Values | Use |
|----------------------|-----------|-----------------------------|--|
| clusterlevel | String | A numeric value (1, 2 or 3) | 1 = Limited, 2 = Typical, 3 = Exhaustive |

The clusterlevel setting determines which methods of cluster key generation are used to generate keys for each driving record. If used, the per-message setting overrides the default setting in the process (that can be adjusted when running a job using the EDQ-CDS Run Profile).

The settings operate as follows:

1 (Limited)

Only cluster keys that do not normally return large numbers of candidate records are generated by the service. This is recommended if working with very large data sets with tight matching requirements.

2 (Typical)

Uses the default methods for generating cluster keys.

3 (Exhaustive)

Methods that may return a large number of candidate records are used to generate cluster keys. This includes methods that only use the name fields. This setting is recommended only when working with low volume data sets (for example, less than a million individuals, or less than 100,000 entities) with loose matching requirements.

4 Matching Services

The Matching services - sometimes referred to as the Match Scoring services in Siebel Universal Data Quality documentation - compare input (driver and candidate) records and produce a list of possible matches, with scores to indicate how good the matches are, and additional information about how the records matched.

In the matching services, the record for comparison is called a driver, and the records it is compared with are known as candidates. Driver records are also compared with each other, but candidate records are never compared with other candidates. Only the highest scoring match for any given record pair is returned.

Note that Siebel currently does not use the Address Matching service, in either batch or real-time, though this service may be integrated with other applications.

4.1 Use

There are three forms of Matching supported:

Real Time

A single driver record (possibly with multiple child entity records) is compared against many candidates.

Full Batch

All records are compared against one another (subject to clustering); for example, all are specified as drivers. This is an extensive operation that can take some time. It is normally used on a new installation, or perhaps as part of a regular maintenance operation.

Incremental Batch

A specific subset of record types are identified and specified as the driver records. Next, other records with matching cluster keys are identified, and specified as the candidates. The driver and candidate records are compared, and the driver records are compared with each other. An example of how this might be used is a regular check on all new records during a set time period, such as a week or month, against pre-existing records.

The real-time matching services are exposed via the following web services in EDQ-CDS:

- IndividualMatch
- EntityMatch
- AddressMatch

The batch and real-time processes that implement the matching services use the following Data Interfaces for input data (mapped to the above web services respectively for real-time matching):

- Individual Candidates
- Entity Candidates
- Address Candidates

The **Matches** data interface is used as a common output interface for all record types (Individual, Entity and Address), although the fields mapped for each record type vary.

4.2 Matching Using Multiple Identifier Values

Matching services within EDQ-CDS are designed to enable users to submit any number of alternative identifier values to use when matching a given individual or entity; for example, multiple email addresses, addresses or names.

EDQ-CDS can perform matching on multiple values of the following attributes if submitted as pipe-delimited lists:

- uid and eid attributes
- alternatephone
- email
- website
- taxnumber
- nationalidnumber
- vatnumber

However, in order for EDQ-CDS to match Individual or Entity records with multiple names or addresses, such records must first be split into multiple records. Each of these records must have the same `entityid` or `individualid`, but with different `nameid` and/or `addressid` attributes.

So, an Individual record with three names must be split into three records, as follows:

| individualid | nameid | firstname | lastname | enail | address1 |
|---------------------|---------------|------------------|-----------------|-------------------|-----------------|
| A1 | 1 | John | Smith | jsmith@jsmith.com | 56 High Street |
| A1 | 2 | Jon | Smith | jsmith@jsmith.com | 56 High Street |
| A1 | 3 | J | Smith | jsmith@jsmith.com | 56 High Street |

Similarly, an Entity record with two addresses must be split into two records:

| entityid | nameid | name | subname | website | address1 |
|-----------------|---------------|-----------------------|------------------|----------------|-----------------|
| B1 | 1 | OracleLtd | Accounts Payable | www.oracle.com | Oracle Parkway |
| B1 | 2 | Oracle Corporation UK | Accounts Payable | www.oracle.com | Oracle Parkway |

And finally, an Entity record with two names and two addresses must be split into four records:

| entityid | nameid | name | subname | website | addressid | address1 |
|-----------------|---------------|-----------------------|------------------|----------------|------------------|--------------------|
| C1 | 1 | OracleLtd | Accounts Payable | www.oracle.com | A | Oracle Parkway |
| C1 | 1 | OracleLtd | Accounts Payable | www.oracle.com | B | Thames Valley Park |
| C1 | 2 | Oracle Corporation UK | Accounts Payable | www.oracle.com | A | Oracle Parkway |
| C1 | 2 | Oracle Corporation UK | Accounts Payable | www.oracle.com | B | Thames Valley Park |

The EDQ Siebel Connector automatically prepares the data to use the matching service appropriately, where EDQ-CDS is integrated with Siebel. If the use of multiple child entities has been configured in Siebel, then the data is prepared in the structure required by the EDQ-CDS matching services. For example, concatenating multiple phone numbers into a pipe-delimited list, and splitting out multiple records if the use of multiple names or addresses is configured.

Note: For records with multiple child entities, only one match will ever be returned between a pair of records. This will always be the highest scoring match according to the match rules.

4.3 Interfaces

The matching web services present input interfaces with direct mappings to the shared 'Candidate' data interfaces as follows:

| Web Service | Input Interface |
|-----------------|---|
| IndividualMatch | See Section 5.1.1, "Individual Candidates." |
| EntityMatch | See Section 5.1.2, "Entity Candidates." |
| AddressMatch | See Section 5.1.3, "Address Candidates." |

All the matching services return output attributes using the common **Matches Interface** data interface.

4.4 Parameters

The IndividualMatch, EntityMatch and AddressMatch web services all expose a matchthreshold attribute, which is used to suppress (not return) matches with a score below this threshold.

| Parameter Attribute | Data Type | Accepted Values | Use |
|---------------------|-----------|-----------------------------------|---|
| matchthreshold | Number | A numeric value between 0 and 100 | Matches that are generated by the matching service with a score below the stated threshold will be suppressed (not returned in the web service response). If used, this overrides the default setting in the process, which can be adjusted when running a job using the EDQ-CDS Run Profile. |
| matchoptions | String | N/A | This parameter is not currently used. It is intended for use in future versions of EDQ-CDS. |

5 Data Interfaces

This section describes the following EDQ-CDS data interfaces:

- [Section 5.1, "Candidate Interfaces"](#)
- [Section 5.2, "Matches Interface"](#)

- [Section 5.3, "Cluster Results Interfaces"](#)

5.1 Candidate Interfaces

The Candidate interfaces are used for data input to individual/entity/address matching and clustering in both batch and real-time.

5.1.1 Individual Candidates

| Attribute Name | Data Type | Supports Multiple Values? (Y/N) | Notes |
|----------------|-----------|---------------------------------|--|
| candidate | String | N | 0 = driving record, 1 = candidate. Used in matching only. All driving records are compared against each other and against each candidate, but candidates are not compared against each other. Note that in real-time matching, there must be one and only one logical driving record representing an Individual, Entity or Address. Multiple physical records for the same Individual or Entity are allowed if multiple child entities are in use. Multiple candidate records may always be presented. |
| individualid | String | N | Unique identifier of the individual (for example, customer, employee, or contact). Mandatory for batch jobs and real time requests containing multiple driver records. |
| languages | String | Y | 3 character Siebel language code. Only used in name standardization to help determine whether a name containing Kanji is Japanese or Chinese. |
| uid1 | String | Y | Unique ID 1 (single or pipe-delimited list of multiple values). Note: The Unique ID fields are used to match records based on custom unique identifiers, such as passport or tax numbers. For more information, see <i>Oracle Enterprise Data Quality Customer Data Services Pack Business Services Guide</i> . |
| uid2 | String | Y | Unique ID 2 (single or pipe-delimited list of multiple values). |
| uid3 | String | Y | Unique ID 3 (single or pipe-delimited list of multiple values). |

| Attribute Name | Data Type | Supports Multiple Values? (Y/N) | Notes |
|------------------|-----------|---------------------------------|--|
| eid1 | String | Y | Elimination ID 1 (single or pipe-delimited list of multiple values). Note: The Elimination ID fields are used to eliminate possible matches between records based on custom unique identifiers, such as passport or tax numbers. For more information, see <i>Oracle Enterprise Data Quality Customer Data Services Pack Business Services Guide</i> . |
| eid2 | String | Y | Elimination ID 2 (single or pipe-delimited list of multiple values). |
| eid3 | String | Y | Elimination ID 3 (single or pipe-delimited list of multiple values). |
| nameid | String | N | Unique identifier for the name, used to distinguish between different names for the same individual when multiple child entities are used. For more information, see Section 4, "Matching Services." |
| title | String | N | Title |
| firstname | String | N | First Name |
| middlename | String | N | Middle Name |
| lastname | String | N | Last Name |
| gender | String | N | Gender (M or F) |
| dob | String | N | Date of Birth, in any of the formats recognized by the *Date Formats reference data set in EDQ. |
| jobtitle | String | N | Job Title |
| homephone | String | N | Home Phone Number |
| workphone | String | N | Work Phone Number |
| mobilephone | String | N | Mobile Phone Number |
| faxphone | String | N | Fax Number |
| alternatephone | String | Y | Alternative Phone Number - either a single value, or a pipe-delimited list of multiple values. |
| email | String | Y | A single value or a pipe-delimited list of multiple email addresses. |
| taxnumber | String | Y | A single value or a pipe-delimited list of multiple tax numbers. |
| nationalidnumber | String | Y | Social Security Number (US) or equivalent, single value or pipe-limited list. |
| accountname | String | N | The name of the account (for example, entity) to which this individual belongs, if relevant. |

| Attribute Name | Data Type | Supports Multiple Values? (Y/N) | Notes |
|-------------------------|-----------|---------------------------------|---|
| addressid | String | N | Unique identifier for the address, used to distinguish between different addresses for the same individual when multiple child entities are used. For more information, see Section 4, "Matching Services." |
| address1 | String | N | Address line 1 |
| address2 | String | N | Address line 2 |
| address3 | String | N | Address line 3 |
| address4 | String | N | Address line 4 |
| dependentlocality | String | N | A smaller population center data element, dependent on the contents of the <code>city</code> field. For example, a Neighborhood in Turkey. For many countries, this attribute is not used. |
| doubledependentlocality | String | N | The smallest population center data element, dependent on both the contents of the <code>city</code> and <code>dependentlocality</code> fields. For example, a village in the UK. For many countries, this attribute is not used. |
| city | String | N | The locality, town or city of the address. |
| subadminarea | String | N | The smallest geographic data element within a country. For example, a county in the USA. |
| adminarea | String | N | The most common geographic data element within a country. For example, USA State or Canadian Province. |
| postalcode | String | N | Postal or zip code for the address, if relevant for the country. Note: With matching services, leading zeroes are stripped only on numeric postalcodes to avoid a numeric postalcode reinterpreted as a number by an external programs where leading zeroes are automatically stripped. For example, Excel may reformat numeric postalcodes as a number by removing the leading zeroes. This is enabled by default in the <code>edq-cds-daas.properties</code> Run Profile. If there are any alpha characters present, the leading zeroes are not stripped. |
| country | String | N | Country name or ISO 2 char code. |
| clusterlevel | String | N | 1 = limited, 2 = typical, 3 = exhaustive. Used in clustering only. If a value is not supplied, the value defined by the override property is used if set; otherwise the default is 2. |

| Attribute Name | Data Type | Supports Multiple Values? (Y/N) | Notes |
|----------------|-----------|---------------------------------|--|
| matchthreshold | Number | N | Minimum match rule score to return a result (0-100). Used in matching only. If a value is not supplied, the value defined by the override property in the Run Profile is used if set; otherwise the default is 70. |
| matchoptions | String | N | For future use. |

5.1.2 Entity Candidates

| Attribute Name | Data Type | Supports Multiple Values? (Y/N) | Notes |
|----------------|-----------|---------------------------------|--|
| candidate | String | N | 0 = driving record, 1 = candidate. Used in matching only. All driving records are compared against each other and against each candidate, but candidates are not compared against each other. |
| entityid | String | N | Unique record identifier. Mandatory for batch jobs and real time requests containing multiple driver records. |
| languages | String | Y | 3 character Siebel language code. Only used in name standardization to help determine whether a name containing Kanji is Japanese or Chinese. |
| uid1 | String | Y | Unique ID 1 (single or pipe-delimited list of multiple values). Note: The Unique ID fields are used to match records based on custom unique identifiers, such as passport or tax numbers. For more information, see <i>Oracle Enterprise Data Quality Customer Data Services Pack Business Services Guide</i> . |
| uid2 | String | Y | Unique ID 2 (single or pipe-delimited list of multiple values). |
| uid3 | String | Y | Unique ID 3 (single or pipe-delimited list of multiple values). |
| eid1 | String | Y | Elimination ID 1 (single or pipe-delimited list of multiple values). Note: The Elimination ID fields are used to eliminate possible matches between records based on custom unique identifiers, such as passport or tax numbers. For more information, see <i>Oracle Enterprise Data Quality Customer Data Services Pack Business Services Guide</i> . |
| eid2 | String | Y | Elimination ID 2 (single or pipe-delimited list of multiple values). |

| Attribute Name | Data Type | Supports Multiple Values? (Y/N) | Notes |
|-------------------------|------------------|--|---|
| eid3 | String | Y | Elimination ID 3 (single or pipe-delimited list of multiple values). |
| nameid | String | N | Unique identifier for the name, used to distinguish between different names for the same entity when multiple child entities are used. For more information, see Section 4, "Matching Services." |
| name | String | N | Organization name, for example, "Oracle Corporation UK". |
| subname | String | N | Department or site, for example, "Reading" or "Accounts Payable". |
| phone | String | N | |
| alternatephone | String | Y | A single or pipe-delimited list of multiple alternative phone number values. |
| website | String | Y | A single or pipe-delimited list of multiple alternative web site addresses. |
| taxnumber | String | Y | A single or pipe-delimited list of multiple tax numbers. |
| vatnumber | String | Y | A single or pipe-delimited list of multiple VAT numbers. |
| addressid | String | N | Unique identifier for the address, used to distinguish between different addresses for the same entity when multiple child entities are used. For more information, see Section 4, "Matching Services." |
| address1 | String | N | Address line 1 |
| address2 | String | N | Address line 2 |
| address3 | String | N | Address line 3 |
| address4 | String | N | Address line 4 |
| dependentlocality | String | N | A smaller population center data element, dependent on the contents of the <code>city</code> field. For example, Turkish Neighborhood. |
| doubledependentlocality | String | N | The smallest population center data element, dependent on both the contents of the <code>city</code> and <code>dependentlocality</code> fields. For example, UK Village. |
| city | String | N | |
| subadminarea | String | N | The smallest geographic data element within a country. For example, USA County. |
| adminarea | String | N | The most common geographic data element within a country. For example, USA State or Canadian Province. |

| Attribute Name | Data Type | Supports Multiple Values? (Y/N) | Notes |
|----------------|-----------|---------------------------------|---|
| postalcode | String | N | Postal or zip code for the address, if relevant for the country. Note: With matching services, leading zeroes are stripped only on numeric postalcodes to avoid a numeric postalcode reinterpreted as a number by an external programs where leading zeroes are automatically stripped. For example, Excel may reformat numeric postalcodes as a number by removing the leading zeroes. This is enabled by default in the <code>edg-cds-daas.properties</code> Run Profile. If there are any alpha characters present, the leading zeroes are not stripped. |
| country | String | N | Country name or ISO 2 char code. |
| clusterlevel | String | N | 1 = limited, 2 = typical, 3 = exhaustive. Used in clustering only. |
| matchthreshold | Number | N | Minimum match rule score to return a result (0-100) . Used in matching only. |
| matchoptions | String | N | For Future Use. |

5.1.3 Address Candidates

| Attribute Name | Data Type | Supports Multiple Values? (Y/N) | Notes |
|-------------------------|-----------|---------------------------------|--|
| candidate | String | N | 0 = driving record, 1 = candidate. Used in matching only. |
| addressid | String | N | Unique identifier for the address. |
| address1 | String | N | Address line 1 |
| address2 | String | N | Address line 2 |
| address3 | String | N | Address line 3 |
| address4 | String | N | Address line 4 |
| dependentlocality | String | N | A smaller population center data element, dependent on the contents of the <code>city</code> field. For example, Turkish Neighborhood. |
| doubledependentlocality | String | N | The smallest population center data element, dependent on both the contents of the <code>city</code> and <code>dependentlocality</code> fields. For example, UK Village. |
| city | String | N | |
| subadminarea | String | N | The smallest geographic data element within a country. For example, USA County. |

| Attribute Name | Data Type | Supports Multiple Values? (Y/N) | Notes |
|----------------|-----------|---------------------------------|--|
| adminarea | String | N | The most common geographic data element within a country. For example, USA State or Canadian Province. |
| postalcode | String | N | Postal or zip code for the address, if relevant for the country. |
| country | String | N | Country name or ISO 2 char code. |
| clusterlevel | String | N | 1 = limited, 2 = typical, 3 = exhaustive. Used in clustering only. |
| matchthreshold | Number | N | Minimum match rule score to return a result (0-100). Used in matching only. |
| matchoptions | String | N | For future use. |

5.2 Matches Interface

The Matches interface is used for the output of the matching services in batch and real-time. It is used for individuals, entities and addresses because it contains no attributes specific to any business object.

With Individual and Entity matching, if there are multiple matches between records with the same `masterid` and `matchids` (for example, due to multiple matches with different names and addresses), only the strongest match (by match score) is returned for the record pair. Siebel does not currently use the returned `masternameid`, `matchnameid`, `masteraddressid` and `matchaddressid` attributes, though these may be used in other integrations to display the correct records in the application according to the best matches.

| Attribute Name | Data Type | Notes |
|-----------------|-----------|--|
| serverid | String | Server ID. Not applicable to Siebel. |
| jobid | String | Job ID. Not applicable to Siebel. |
| masterid | String | Driving record ID. Only used in Batch. |
| matchid | String | Matching record ID. |
| masternameid | String | Driving record name ID. Used to identify which driving record <code>nameid</code> matched, where multiple names were presented. |
| matchnameid | String | Matching record name ID. Used to identify which candidate record <code>nameid</code> matched, where multiple names were presented. |
| masteraddressid | String | Driving record address ID. Used to identify which driving record <code>addressid</code> matched, where multiple addresses were presented. |
| matchaddressid | String | Matching record address ID. Used to identify which candidate record <code>addressid</code> matched, where multiple addresses were presented. |
| matchscore | Number | Match score. |
| rulename | String | Match rule name. |

| Attribute Name | Data Type | Notes |
|-------------------|-----------|--|
| reversedriverflag | String | A flag indicating that an additional, reversed match record has been generated where there is a match between driving records in Batch matching. Valid values are Y and N. |

Note:

- In Siebel integrations, the driving record(s) are also returned in the output from real-time matching requests, with a blank match score and rule name. This behavior is controlled by the phase.*.process.*.Return\ Real-time\ Driving\ Record Run Profile property, and therefore could be configured for other types of integration if required.
 - So that external applications such as Siebel can simply consume the output from batch matching to update both records in a match, CDS batch matching provides two records for each match between driving records. Therefore, if A matches B, a record is returned with masterid A and matchid B, *and* an additional record is generated and returned with masterid B and matchid A. This additionally generated record will have reversedriverflag set to Y in case the external application does not need the additionally generated record.
-
-

5.3 Cluster Results Interfaces

Two data interfaces are used for the output of the results of clustering; one for batch and one for real-time. They are used for entities, individuals, and addresses as they contain no attributes specific to a particular business object. The batch and real-time interfaces contain similar information, with the main difference being the way in which the results are processed. The Batch and Real-Time Results Interfaces contain similar information, the main difference is the way in which the results are processed.

5.3.1 Real-Time Cluster Results Interface

The Real-time Cluster Results interface is used for the output of Clustering Services in real-time. The output values are returned in arrays in no specific order; the `clustervalues` and `clusterlevels` array element always correspond.

| Attribute Name | Data Type | Notes |
|----------------|-------------|--|
| externalid | String | ID of the individual, entity or address of the clustered record. |
| clustervalues | StringArray | Cluster key value array. |
| clusterlevels | StringArray | Cluster key level array. |

5.3.2 Batch Cluster Results Interface

The Batch Cluster Results interface is only used in the Batch Clustering service. It differs from the Real-time Cluster Results interface in that it returns one row per cluster value per record, rather than arrays of cluster values for a record.

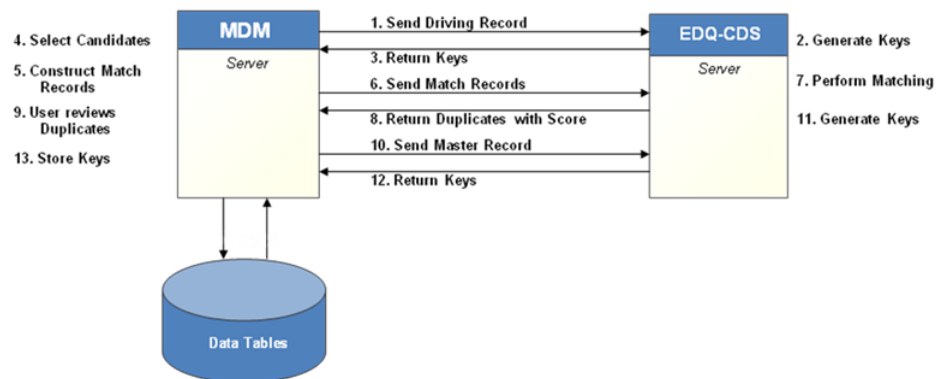
| Attribute Name | Data Type | Notes |
|----------------|-----------|---|
| serverid | String | Server ID. Not applicable to Siebel. |
| jobid | String | Job ID. Not applicable to Siebel. |
| externalid | String | ID of the individual, entity, or address of the clustered record. |
| clustervalue | String | Cluster key value. |
| clusterlevel | String | Cluster key level. |

6 Real-Time Integration

The EDQ-CDS real-time matching services can be called by an external application without any changes to the default configuration. It is the responsibility of the calling application to manage the storage of record cluster keys and to perform the selection of match candidates to be passed to the matching service.

A typical interaction between the calling application (for example, a CRM or Master Data Management [MDM] application) and EDQ-CDS during real-time matching (for example, Contact duplicate prevention) is illustrated as follows:

Figure 1 Overview of Expected Integration Architecture with Matching Services



In detail the matching services operate as follows:

- 1. Send Driving Record** — The application sends the new (driving) record and the configured cluster level to EDQ-CDS.
- 2. Generate Keys** — EDQ-CDS generates the cluster key(s) for the driving record.
- 3. Return Keys** — EDQ-CDS returns the driving record's cluster keys to the MDM application.
- 4. Select Candidates** — The MDM application selects all (candidate) records that share any of the same cluster keys. If no candidates are identified then go to step 13.
- 5. Construct Match Data** — The MDM application constructs the match data for the driving and candidate records
- 6. Send Match Records** — The MDM application sends the data for the new (driving) record and candidates to EDQ-CDS.

7. **Perform Matching** — EDQ-CDS matches the driving record against the candidates to identify potential duplicates. Each match is assigned a score indicating the strength of match.
8. **Return Duplicates with score** — EDQ-CDS returns the IDs of the matched candidates (and scores) to the MDM application. The driving record is also returned, but with a blank score. If no duplicates were identified by EDQ-CDS then go to step 13.
9. **User reviews Duplicates** — As indicated.
10. **Send Master Record** — If duplicates were identified by EDQ-CDS and selected by the user, then the driving record is merged with the existing duplicate record. If a merge operation occurred then the MDM application sends the new merged (master) record details back to EDQ-CDS.
11. **Generate Keys** — EDQ-CDS uses the details of the master record to generate cluster key values.
12. **Return Keys** — EDQ-CDS returns the master record's cluster keys to the MDM application.
13. **Store Keys** — The MDM application stores the cluster keys for new master record.

7 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

8 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.

Oracle Enterprise Data Quality, Release 11g R1 (11.1.1.7)
E40733-02

Copyright © 2006, 2013, 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.