Oracle® Enterprise Data Quality

Customer Data Services Pack Matching Guide

Release 11g R1 (11.1.1.7)

E40737-01

October 2013

1 Introduction

Oracle Enterprise Data Quality Customer Data Services Pack (EDQ-CDS) has been designed to match customer data that exhibits real-world variability. All relevant matches in the data set are presented back and appropriately scored according to the likelihood of a match between records. To do this, it uses a variety of different mechanisms, including the application of a wide range of matching algorithms on the data as it is presented, as well as matching techniques on derived forms of the data.

For example, names presented in one writing system are matched both using this writing system and also using a transformed version of the name, providing effective cross-script matching. Similarly, addresses are matched in near raw form (after standardization of international address words and phrases, and after removal of filler words), but also by extracting and matching key information from the address, such as the likely building number, sub-building number, and postal code.

1.1 Objectives of Matching

In general, the matching services provided by EDQ-CDS are designed for duplicate prevention, rather than searching. This means that the intention of the out-of-the-box services is to intervene when a record is added to a system if it appears that it may already exist. The implication of this is that the matching services are focused on much more than a single attribute (such as Name) and deliberately do not cast as wide a net as a typical search operation. There may be other records in the system that are not matched but which have similar details, perhaps even exactly the same name, but where the secondary identification information indicates that a match is unlikely. In these cases, EDQ-CDS aims to minimize the additional work for users or data stewards whose role it is to resolve possible matches. This makes the product ideally suited to operate as the data quality protection component of a Master Data Management system, such as Oracle Customer Hub, where the purpose of the services is to link as many records as possible together automatically with as little noise as possible. The same is true for a Customer Relationship Management system, such as Siebel.

Note : It is possible to change the configuration of EDQ-CDS in order to perform more exhaustive matching. This is mainly designed for use with low volume, high value data sets that do not necessarily offer sufficient secondary information (beyond name fields).

1.2 Multiple Locales and Languages

EDQ-CDS has been designed as a multi-locale system, and uses international and culture-sensitive name transcription, transliteration and variant recognition techniques, as well as using international dictionaries when standardizing and matching addresses.

The system is designed to work with international data, and provides international dictionaries of name and address standardizations for this purpose. The international 'Latin script' dictionaries provide coverage of the following 'base' locales, amongst others:

- United States and Canada
- United Kingdom
- France
- Germany
- Italy
- Spain
- Portugal
- Brazil
- Greece
- Ireland
- Austria
- Turkey
- South Africa
- Australia and New Zealand
- Scandinavia
- Argentina
- Mexico

In addition to these base locales, EDQ-CDS provides specific optional capabilities for advanced handling of data from the following locales:

- Arab World (Arabic and Mixed Arabic/Latin)
- Japan (Kanji, Katakana and Hiragana)
- China (Simplified and Traditional Chinese)
- Russia
- Korea (Hangul)

The set of enabled languages is determined by the configuration of the EDQ-CDS - Initialize Reference Data project, so that the same reference data may be used by any number of EDQ-CDS matching servers. By default, reference data sets for the base locales are pre-initialized in the EDQ server landing area, but these can be easily overwritten either by unzipping `cdslists-initialized-full.zip` over these files (to provided coverage for all supported locales and languages) or by configuring and running the Initialization job.

1.3 Uses of Matching

The Matching processes included in EDQ-CDS are designed primarily for the following use cases:

- Duplicate Prevention - uses the Cluster Generation and Matching web services to prevent duplicate records being entered into applications.
- Regular Batch Matching for Duplicate Removal - uses the Batch Matching job, run on all, or a subset of, data in an application, and links records together for potential merge.

It is also possible to use the Batch Matching processes as a template for the deduplication of records before they are loaded into a system. This is likely to require additional configuration, and use of EDQ. In such circumstances the best practice is to understand the data before matching using data profiling and audit techniques, such as those available in the EDQ-CDS Data Quality Health Check. In most cases, the set of enabled match rules will need some tuning towards the specifics of the in-scope data in order to provide the optimum balance between performance and effectiveness. It may also be necessary to use EDQ's Match Review application to review possible matches, and construct rules for merging records together.

Note : EDQ-CDS does not provide any out-of-the-box merging (or 'survivorship' configuration, because in the two main use cases, merging is performed by the calling application after matches have been identified.

1.3.1 Duplicate Prevention

EDQ-CDS uses stateless web services for duplicate prevention to avoid complex replication and synchronization of large volume customer data. This places the following requirements on the application integrating with EDQ:

1. Storage of Cluster Key tables for each type of record (for example, Contacts or Accounts). These are normally thin tables with two columns - the Primary Key of the record and the Cluster Key. The table must allow for multiple key values per record.
2. Functionality to select and construct candidate records to submit to the Matching service. This involves:
    a. Querying the Cluster Key table for the relevant record, and finding all records that share a key value with the driving record.
    b. Constructing the data that is required for matching for each of these records.
    c. Submitting these Candidate records together with the driving record to the Matching service.

Optimum Duplicate Prevention Process Flow

In order to access the full capabilities of EDQ-CDS for duplicate prevention, the integration should work as follows:

1. To prepare the system for real-time duplicate prevention, key values are generated for each record in Batch using the Cluster Key Generation process. This can occur either when migrating the data into the application, or as a batch process to generate the key values into the application's Cluster Key tables.

2. When a record is added or updated in the application, the Cluster Key Generation service is called in real-time, and returns a number of cluster key values for the record.
3. The application then selects candidate records (those records which share a common key with the driving record) using the existing stored keys and submits them along with the driving record to the Matching service.
4. The Matching service decides which of the candidates are a likely match to the driving record and returns the ids of these records, and a score indicating the strength of match.
5. The application then decides how to consume the matching results; for example, whether to 'auto-match' or present possible matches to the user so that a decision can be made whether or not to continue with inserting a record, or merge it with an existing record.
6. If the record is merged with another record to create a changed master record, an additional call should be made to the Cluster Generation service in order to re-generate the correct cluster key values before committing the record.

In this model, complex multi-locale EDQ techniques are used to generate the keys and ensure that the right balance between performance and matching effectiveness is maintained, while ensuring that the calling application retains control of data integrity and transactional commits.

1.3.2 Batch Matching

When working with Siebel CRM, Siebel's Data Quality Manager is used to instigate batch jobs and a shared staging database is used to write records for matching and to consume match results. The EDQ-CDS batch matching processes automatically adjust to Siebel's 'Full Match' (match all records against each other) and 'Incremental Match' (match a subset of records against all of their selected candidates) modes.

1.4 Match Tuning

In EDQ-CDS matching, it is not necessary to be overly concerned with which identifiers will be populated in the data that is worked with. EDQ-CDS does not use a weighted algorithm that will place unnecessary emphasis on unpopulated data, and so does not require adjustment for this.

Matching works by examining all of the available data and attempting various ways to form a match. The matching design builds in the knowledge of how strong an identifier is likely to be based on real world principles. A significant advantage of this approach is that there is normally no need to apply algorithmic tuning adjustments, the results of which are hard to predict. Instead, match tuning is normally a matter of performing one of the following tasks, which will have a more predictable effect on match results:

- Adjusting the clustering configuration.
- Enabling or disabling a provided rule.
- Adjusting the score that a specific rule returns.
- Inserting a new rule (perhaps a stronger or weaker version of an existing rule).

Note:

- Even when inserting a new rule, it may well be possible to use existing comparisons and comparison results rather than adding new comparisons, though both are possible.
- For batch matching of large data sets, it is recommended that redundant match rules [whose priority score is lower than the `matchthreshold` setting] are disabled as this will yield significant performance improvements.

2 Clustering

2.1 Concepts

Clustering is used to minimize the work that is performed during the final stage of matching. It works by splitting the records into tranches (clusters), based on similarities in significant data fields. Only subsets of the data which share similar characteristics (and will therefore be placed in the same cluster) will be compared on a record-by-record basis during matching.

If loose clusters are used, there will be a large number of records in each cluster. This means that there is a reduced risk that true matches will be missed, but also that a greater amount of processing will be required to compare all the clustered records. It will also increase the number of false positives being returned, which will require extra time to assess. A tighter clustering strategy will result in smaller cluster groups and hence a reduced processing time, but will increase the likelihood that some true matches will not be detected.

EDQ-CDS is supplied with a number of different clustering algorithms for individual and entity data that use different combinations of key data fields in their construction. Each clustering algorithm has been assigned a unique prefix code for easy identification, and to ensure keys from different clusters are not identical. This prefix and all data elements within a cluster value are separated with the caret symbol (^).

2.2 Cluster Level

All clustering algorithms are assigned a cluster level which relates to the tightness of the cluster groups that it generates with typical data. The following cluster level settings are available:

Level	Name	Usage
1	Limited	Useful for tight matching with large volumes of data.
2	Typical	The recommended setting for most applications providing a balance of performance with match tolerance.
3	Exhaustive	Required for the loosest possible matching where there is high risk if matches are missed.

2.3 Cluster Values

The format of the cluster values is:

```
[Prefix]^[Cluster Level]^[Cluster Value]
```

Additional components are further delimited with the ^ symbol:

```
[Prefix]^[Cluster Level]^[Cluster Value 1]^[Cluster Value 2]
```

2.4 Individual Clustering

The following clustering algorithms are provided for matching individual data:

Prefix	Cluster Name	Level	Description
LMP	Family Name Meta, Postal Code	1	4-character double-metaphone of the surname + First 5 characters of the postal code + First 3 characters of address1. **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 `edq-cds-daas.properties` Run Profile. If there are any alpha characters present, the leading zeroes are not stripped.
PLN	Phone last N	1	Last N digits of the phone/fax/work/mobile number. **Note**: The default value is 6. This can be changed in the EDQ-CDS Run Profile.
EF9	Email first 9	1	First 9 characters of the email address.
TAX	Tax Number	1	First 10 characters of the tax number.
EID1 EID2 EID3	Elimination Identifier	1	All non-alphanumeric characters are removed.
UID1 UID2 UID3	Unique Identifier	1	All non-alphanumeric characters are removed.
NID	National Identifier	1	First 10 characters of the National ID number.
FLP	Given Names standardized, Family Name, Postal Code	2	First character of the standardized given name + First 3 characters of the family name + First 5 characters of the postal code.
FLY	Given Names standardized, Family Name, City	2	First 3 characters of the standardized given name + First 3 characters of the family name + First 10 characters of the city name.
FA1	Given Names standardized, Address1	2	First 3 characters of the standardized given name + First 3 characters of the family name + First 10 characters of address line 1.
LMC	Family Name Meta, First Company word	2	First 4 characters of the family name + First word of the account name.
A5F	Address1, Address2, City	3	First 5 characters of address line 1 + First 5 characters of address line 2 + First 5 characters of the city name.

Prefix	Cluster Name	Level	Description
OSP	Original Script name, Postal Code	3	First 4 characters of the original script name + First 4 characters of the postal code.
FLM	Full Name Meta	3	The full name tokens are sorted and then the double-metaphone algorithm is applied to generate tokens of up to 3 characters in length. For each ordered pair of tokens, a cluster value is generated that is the concatenation of the two metaphone tokens.

Note : The clustering algorithms use data attributes that have been normalized (for example, converted to upper case and symbols stripped) and have had whitespace removed. This allows clustering and matching to be performed in a case-insensitive manner and to be tolerant of the spacing within attributes.

2.4.1 Examples

The following record data is used to provide examples of the cluster values that are generated by the individual clustering algorithms:

Attribute	Value
firstname	Jim
middlename	Frederick
lastname	Smith
mobilephone	077777 123456
email	j.smith@mymail.com
taxnumber	888666444
accountname	Acme Ltd
address1	14 high St
city	Cambridge
postalcode	CB1 2AB
uid1	00021-53563
eid1	gbr0008873323
nationalidnumber	AB 12 34 56 C

The cluster values that are generated using a `clusterlevel` setting of `3` (Exhaustive) are as follows:

Cluster Prefix	Cluster Values
LMP	LMP^1^SM0^CB12A^14H
PLN	PLN^1^123456
EF9	EF9^1^J.SMITH@M

Cluster Prefix	Cluster Values
TAX	TAX^1^888666444
EID1	EID1^1^GBR0008873323
UID1	UID1^1^0002153563
NID	NID^1^AB123456C
FLP	FLP^2^J^SMI^CB12A
FLY	FLY^2^JAM^SMI^CAMBRIDGE
FA1	FA1^2^JAM^14HIGH
LMC	LMC^2^SM0^ACME
A5F	A5F^3^14HIG^^CAMBR
FLM	FLM^3^FRTJMS FLM^3^FRTSM0 FLM^3^JMSSM0

2.5 Entity Clustering

The following clustering algorithms are provided for matching entity data:

Prefix	Cluster Name	Level	Description
APC	Address 1 and Postal Code	1	First 3 characters of address line 1 + First 5 characters of the postal code. **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 edq-cds-daas.properties Run Profile. If there are any alpha characters present, the leading zeroes are not stripped.
TAX	Tax Number	1	First 10 characters of the tax number.
VAT	VAT Number	1	First 10 characters of the VAT number.
PLN	Phone Last N Digits	1	Last N digits of the phone/fax/work/mobile number. **Note**: The default value is 6. This can be changed in the EDQ-CDS Run Profile.
NSD	Name and Sub-name	1	First 30 characters of the concatenation of the distilled name and sub-name.
EID1 EID2 EID3	Elimination Identifier	1	All non-alphanumeric characters are removed.
UID1 UID2 UID3	Unique Identifier	1	All non-alphanumeric characters are removed.
NPC	Name and Postal Code	2	First 4 characters of the name + First 3 characters of the postal code.

Prefix	Cluster Name	Level	Description
NMP	Name Metaphone, Address 1 and Postal Code	2	For each token in the distilled name: 4-character double metaphone of the token + First 4 characters of address line 1 + First 3 characters of the postal code.
WS	Website Stem	2	Website address without the top level domain name, common address prefix and any page portion of the url.
NMA	Full Name metaphone, Address No Numbers	2	Full name double-metaphone 4: address lines 1-4, concatenated, number words stripped, denoised including hyphens, first 10 characters.
NSM	Name metaphone and Sub-name metaphone	3	4-character double-metaphone of the name + 4-character double-metaphone of the sub-name.
OS	Original Script	3	For each token in the original script name: First 5 characters of the name token. For Chinese, Japanese and Korean script each token will generate a cluster value.
NST	Name and Sub-name Tokens	3	Generate a cluster value for the 4-character double metaphone of each token in the distilled name and distilled sub-name.

Note : The clustering algorithms use data attributes that have been normalized (for example, converted to upper case and symbols stripped) and whitespace removed. This allows clustering and matching to be performed in a case-insensitive manner and be tolerant to the spacing within attributes.

2.5.1 Examples

The following record data is used to provide examples of the cluster values that are generated by the entity clustering algorithms:

Attribute	Value
name	Oracle UK
subname	Cambridge
phone	+441223228400
website	http://www.oracle.com/uk
taxnumber	RGW432D243224
vatnumber	999111
address1	296 Cambridge Science Park
city	Cambridge
postalcode	CB4 0WD
uid1	00021-53563
eid1	gbr0008873323

The cluster values that are generated using a `clusterlevel` setting of `3` (Exhaustive) are as follows:

Cluster Prefix	Cluster Values
APC	APC^1^296^CB40W
TAX	TAX^1^RGW432D243
VAT	VAT^1^999111
PLN	PLN^1^228400
NSD	NSD^1^ORACLECAMBRIDGE
EID1	EID1^1^GBR0008873323
UID1	UID1^1^0002153563
NPC	NPC^2^ORAC^CB4
NMP	NMP^2^ARKL^296C^CB4
WS	WS^2^ORACLE
NMA	NMA^2^ARKL^CAMBRIDGES
NSM	NSM^3^ARKL^KMPR
NST	NST^3^ARKL
	NST^3^KMPR

2.6 Address Clustering

The following clustering algorithms are provided for matching address data:

Prefix	Cluster Name	Level	Description
PPC	Premise and Postal Code	1	Premise, first number word, or if no number word first 8 of premise. If no premise first 8 of address1 + Postal code first 5, if no postal code, first 8 of city. **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 `edq-cds-daas.properties` Run Profile. If there are any alpha characters present, the leading zeroes are not stripped.
PC	Postal Code	3	PostalCode, whole value.
A12	Address1 and Address2	2	Address1 distilled, first 10. Address2 distilled, first 10.
A1C	Address1 and City	2	Address1 distilled, first 5. City, First 8.
FA	Full Address	1	Full Address distilled, first 12. Cluster not generated if there are fewer than 12 characters.
FAN	Full Address No Number Words	2	Address lines 1-4, concatenated, number words stripped, first 10. Cluster not generated if there are fewer than 10 characters.

Note:

- A **Number word** is a word with one or more numbers within it. for example, 234 and 2A are both number words.
- The clustering algorithms use data attributes that have been normalized (for example, converted to upper case and symbols stripped) and whitespace removed. This allows clustering and matching to be performed in a case-insensitive manner and be tolerant to the spacing within attributes.

2.6.1 Examples

The following record data is used to provide examples of the cluster values that are generated by the address clustering algorithms:

Attribute	Value
address1	2529 CINCINNATI ST
address2	APT 6
city	LOS ANGELES
adminarea	CA
postalcode	90033

Note : During Cluster Key generation, ST is distilled out of the address1 field, and APT is distilled out of the address2 field. This is because they are common addressing components that are less important identifiers than the remainder of the address line, and removing them produces more accurate clusters.

The cluster values that are generated using a clusterlevel setting of 3 (Exhaustive) are as follows:

Cluster Prefix	Cluster Values
PPC	PPC^1^2529^90033
PC	PC^3^90033
A12	A12^2^2529CINCIN^6
A1C	A1C^2^2529C^LOSANGEL
FA	FA^1^2529CINCINNA
FAN	FAN^2^CINCINNATI

3 Individual Matching

The matching design for individuals in CDS is based on Name as the primary identifier for individuals, purely because it should always be present, rather than because it is a strong identifier. However, in general, the aim of the services is only to return matches where the Name matches (using one of a wide variety of matching techniques) and at least one other secondary identifier (such as Email Address,

Address, Date of Birth, any Phone Number, or Social Security Number) also matches (again using a variety of techniques).

In large data sets, there are likely to be a large number of individuals with the same or similar names, but if none of the secondary information matches, it is highly unlikely to be the same person. Even if the secondary information is unpopulated on one or both records, and a match is a little more likely in theory, the absence of the information makes it nearly impossible for a user or data steward to determine if the individual is the same even if in direct contact with the individual. For this reason, matches such as this are not considered using the default rules.

However, matches where only one of the secondary identifiers match (for example, where an email address matches but the address is entirely different) are presented, and offer a strong route to improved data quality, as it is very likely to be the same person (they could, for example, have simply moved house).

3.1 Individual Name Matching

The rules for matching individual names include the use of pre-matching transformations and various matching comparisons in order to handle the following types of variance between different representations of what may be the same individual name:

- Names written in different writing systems/scripts, for example, 'Зоран' and 'Zoran'.
- Variants of the same name, for example, 'Bill' and 'William'.
- Different levels of name completeness, for example, 'Joseph Andrew Harris' and 'Joseph Harris'.
- Name tokens in a different order, for example, 'Lacazette Jacques' and 'Jacques Lacazette'.
- Abbreviated forms of names, for example, 'Chris' and 'Christian'.
- Typographic differences, for example, 'Michael' and 'Micheal'.
- The use of initials, for example, 'A M' and 'Alexander Martin'.
- Changes of surname due to marriage, for example, 'Paula Jones' and 'Paula Lewis' at the same address.
- Various combinations of the above types of variance.

The match rules are organized into groups of rules where all rules in each group have the same name matching rule, but different rules on secondary identifiers (such as address, email address, phone number and so on). The following table lists all of the groups, and therefore all of the name matching rules used.

> **Note :** In this table the pipe character is used to indicate a separator between the input given name and family name attributes (for example, Given Name= Martin, Family Name=Smith is written as 'Martin|Smith'). Where no pipe character is used, this means the Full Name is used in the match rule.

Name Matching Rule	Example Name Match
Script full name exact	Зоран Александрович Макаров =Зоран Александрович Макаров
Name exact	Martin | Fox = Martin | Fox
Standardized given name	Bill | Lewis = William | Lewis
Given name abbreviated	Chris | Smith = Christina | Smith
Standardized given name abbreviated	Abell | Hernandez = Abelson | Hernandez
Script full name any order	Макаров Зоран Александрович =Зоран Александрович Макаров
Given name similar and sounds like	Yngrid | Martin = Ingrid | Martin
First name similar and sounds like	Yngrid Elisabeth | Martin = Ingrid Martin
Additional given names	Michael John | Smith = John | Smith
Standardized full name	Mehmood Mahomed = Mahmoud Mohammed
Script full name has additional names	Зоран Макаров =Зоран Александрович Макаров
Additional names	Mary Jones Steward = Mary Jones
Script full name typos	Зоран Александрович Макаров =Зоран Александрович Маккаров
Standardized given name abbreviated; family name typos	Abell | Hernandez = Abelson | Hernandes
Full name typos, all words	Mary Cloire Jonez = Mary Claire Jones
First name first three; family name typos	Ros Susan | Jonez = Rose Susan | Jones
Full name initials in order; additional names	G A | Smith = Gordon Alfred | Smith
Standardized first name only; female	Jacklin | Jones = Jacqueline | Smith

3.2 Individual Secondary Identifier Matching

For each individual name match rule, and therefore within each match rule group, a number of match rules exist, each with different levels of matching on secondary identifiers, such as Company Name, Email Address, Address, Date of Birth, and phone numbers.

The following table is a guide to the criteria needed to match on each rule. These criteria are combined with the name matching rule in order to determine which match rule is hit, and therefore the score of the match.

Note:

- All matching on secondary identifiers uses prepared versions of the secondary identifiers; for example, all address match rules are applied on prepared versions of the addresses, after various word and phrase standardizations are applied.
- A rule is not included for every combination of secondary identifiers matching; for example, there is no rule that requires a match on *both* Date of Birth and Phone number, as both of the identifiers are suitably strong that even if only one of the attributes match, the match should be generated and scored highly.

Secondary Identifier Match Rule	Description
DOB; e-mail	Date of birth and e-mail match exactly.
Address; e-mail	Address and e-mail match exactly.
E-mail; phone number	E-mail and any phone number match exactly.
Company; address	All tokens in the shorter company name match in the longer company name, and the address matches exactly.
Tax number	Tax number matches exactly.
National ID number	National ID number matches exactly.
E-mail	E-mail matches exactly.
Address	Address matches exactly.
Phone	Any phone number matches exactly.
Premise; subpremise; postal code starts with	Address matches by extracted premise, subpremise and postal code **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 `edq-cds-daas.properties` Run Profile. If there are any alpha characters present, the leading zeroes are not stripped.
Premise; no subpremise; postal code starts with	Address matches by extracted premise and postal code, and there is no data in either `subpremise` field.
DOB	Date of birth matches exactly.
Phone last N digits	Any phone number matches using the last N digits (tby default, the last 6 digits.)
Company; postal code	All tokens in the shorter company name match in the longer company name, and the postal code matches exactly.
Address all words	All words in the shorter address match in the longer address.

Secondary Identifier Match Rule	Description
DOB similar	Dates of birth are a close match (a day/month transposition match using the default comparison settings).
Tax number typos	Tax number matches with a Character Edit Distance of 1 or 2.
National ID number typos	National ID number matches with a Character Edit Distance of 1 or 2.
E-mail typos	E-mail matches with a Character Edit Distance of 1 or 2.
Address all words typos	All words in the shorter address match in the longer address with a Character Error Tolerance of 20%.
Address similar; postal code	Address matches with a Character Match Percentage of 65 or more, and the postal code matches exactly.
Address similar; first address one word	Address matches with a Character Match Percentage of 65 or more, and there is at least one token match in the first line of the address.
Company	All tokens in the shorter company name match in the longer company name.

It is also possible to perform matching or elimination of Individual records using custom unique identifiers, see Section 5, "ID Matching."

4 Entity Matching

As with individuals, the design for EDQ-CDS Entity matching is based around the name, but with acknowledgement that the name is a less strong identifier in the context of an Entity, as Entities change name more frequently than individuals. Also, there tends to be less secondary information on Entity records. As a result, Entity matching is based largely on Name and Location (Address) attributes, though matching on additional identifiers such as URLs and Tax Numbers is also provided.

> **Note :** It is significantly harder to match entities (as opposed to individuals) between different writing systems, as the process of transliteration — and even transcription — is much less likely to be successful. Very often, the only way to recognize that a company is the same when written in two different languages is to hold huge dictionaries of all possible company names and their appropriate translations (rather than transliterations or transcriptions). In most cases, such data is simply not available though if it is available it can be plugged into EDQ-CDS in order to improve results.

4.1 Entity Name Matching

The rules for matching entity names include the use of pre-matching transformations and various matching comparisons in order to handle the following types of variance between different representations of what may be the same entity name:

- Entity names written in different writing systems.
- Entity names with or without suffixes, for example, 'Oracle LTD' and 'Oracle'.

- Entity names containing abbreviated terms or suffixes, for example, 'Oracle Limited' and 'Oracle LTD'.
- Character order and spelling differences/errors in entity names, for example, 'Oracle' and 'Oralce'.
- Entity names with different levels of name completeness, for example, 'ABC Technology Consultants LTD' and 'ABC Technology LTD'.
- Entity name tokens appearing in a different order, for example, 'Cambridge Science Park LTD' and 'Science Park Cambridge'.
- Potential matches where there is no name match at all but strongly matching secondary identifiers (for example, if a company has been renamed there may be two records with identical VAT numbers).

The match rules are organized into groups of rules where all rules in each group have the same name matching rule, but different rules on secondary identifiers (such as address, or URL). The following table lists all of the groups, and therefore all of the entity name matching rules used.

> **Note :** In the following table, where a name matching rule uses the 'full name', this means it applies to the entity full name identifier, a concatenation of the entity name and sub-name attributes. The pipe (|) character is used to separate the entity name and sub-name were the sub-name attribute is required to provide an example match.

Entity Name Matching Rule	Example Entity Name Match
Script full name exact	ДИРЕКЦИЯ БАЛТ-Й АЭС = ДИРЕКЦИЯ БАЛТ-Й АЭС
Full name exact	TCHIBO GMBH = TCHIBO GMBH
Standardized full name exact	ORACLE UK LTD | READING = ORACLE UK LIMITED | READING
Script full name without suffixes exact	Открываем частное образовательное учреждение = Открываем частное образовательное
Full name without suffixes exact	ORACLE = ORACLE CORPORATION
Full name without suffixes similar and sounds like	ORACLE CAMBRIDGE SCIENCE PARK = ORACLE CAMBRIDGE PARK SCIENCE
Script full name out of order	ДИРЕКЦИЯ БАЛТ-Й АЭС = ДИРЕКЦИЯ АЭС БАЛТ-Й
Script full name without suffixes all words out of order	ОТКРЫВАЕМ ЧАСТНОЕ = ОТКРЫВАЕМ ОБРАЗОВАТЕЛЬНОЕ
Full name without suffixes all words out of order	CAMBRIDGE SCIENCE PARK LTD = SCIENCE PARK CAMBRIDGE
Script full name has additional names	Открываем частное учреждение | Москва = Открываем частное образовательное учреждение | Москва
Script entity name without suffixes exact	ОТКРЫВАЕМ ЧАСТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ |МОСКВА = ОТКРЫВАЕМ ЧАСТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ | Колпино

Entity Name Matching Rule	Example Entity Name Match
Entity name without suffixes exact	ORACLE CORPORATION | CAMBRIDGE = ORACLE | READING
Full name all words shorter with typos	Oracle Inc | Cambridge =Oracl | Cambridge
Script entity name without suffixes starts with	ОТКРЫВАЕМ ЧАСТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ| МОСКВА = ОТКРЫВАЕМ ЧАСТНОЕ УЧРЕЖДЕНИЕ| Колпино
Entity name without suffixes starts with	ABC TECHNOLOGY CONSULTANTS LTD = ABC TECHNOLOGY LTD
Script full name without suffixes all words shorter with typos	ОТКРЫВАЕМ ЧАСТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ= ОТКРЫВАЕМ ЧАСТНОЕБ
Full name without suffixes all words shorter with typos	Federal Mogull | Camshafts Inc = Federal Mogul Camshafts Castings Ltd
Script full name typos	ОТКРЫВАЕМ ЧАСТНОЕ УЧРЕЖДЕНИЕ| МОСКВА = ОТКРЫ ЧАСТНОЕ УЧРЕЖДЕНИЕ| МОСКВА
Full name typos	ABD SERVICES LTD = ABC SERVICES LTD
Script full name without suffixes typos	ОТКРЫВАЕМ ЧАСТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ = ОТКРЫВА ЧАСТНОЕ ОБРАЗОВАТЕЛЬНОЕ
Full name without suffixes typos	ABD ENGINEERING LTD = ABC ENGINEERING
Script entity name without suffixes starts with	ОТКРЫВАЕМ ЧАСТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ = УЧРЕЖДЕНИЕ ОТКРЫВАЕМ
Entity name without suffixes starts with	ABC LIMITED| CAMBRIDGE = ABC PHARMACEUTICALS LIMITED| READING
Non-name rules	N/A - These rules are used in order to raise matches where only the secondary data (such as VAT number) matches.

4.2 Entity Secondary Identifier Matching

For each Entity Name match rule, and therefore within each match rule group, a number of match rules exist. Each has different levels of matching on secondary identifiers, such as Address, Website Address, Tax Number, VAT Number or Phone Number.

The following table is a guide to the criteria needed to match on each rule. These criteria are combined with the entity name matching rule in order to determine which match rule is triggered, and therefore the score of the match.

Note:

- All matching on secondary identifiers uses prepared versions of the secondary identifiers; for example, all address match rules are applied on prepared versions of the addresses, after various word and phrase standardizations are applied.
- A rule is not included for every combination of secondary identifiers matching - for example, there is no rule that requires a match on **both** Tax Number and VAT Number, as both of the identifiers are suitably strong that even if only one of the attributes match, the match should be generated and scored highly.

Secondary Identifier Match Rule	Description
Address	Address matches exactly.
Premise; subpremise; postal code starts with	Address matches by extracted premise, subpremise and postal code. **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. If there are any alpha characters present, the leading zeroes are not stripped.
Premise; no subpremise; postal code starts with	Address matches by extracted premise and postal code, and there is no data in either `subpremise` field.
Address all words	All words in the shorter address match in the longer address.
Address all words typos	All words in the shorter address match in the longer address with a Character Error Tolerance of 20%.
Website; phone number	The website address and any phone number match exactly.
Tax number	The tax number matches exactly.
VAT number	The VAT number matches exactly.
Address 1 typo; city; country	The address is similar and both the city and country matches exactly.
Address similar; postal code	Address matches with a Character Match Percentage of 65 or more, and the postal code matches exactly.
Phone	Any phone number matches exactly.
Phone last N digits	Any phone number matches using the last N digits (by default, the last 6 digits.)
Tax number typos	The tax number matches with a Character Edit Distance of 1 or 2.
VAT number typos	The VAT number matches with a Character Edit Distance of 1 or 2.
Postal code	The postal code matches exactly.

Secondary Identifier Match Rule	Description
City; country	The city and country match exactly.
Website	The website address matches exactly.
Website stem	The stem part of the website address matches exactly.
City	The full city name matches exactly.
Address similar; first address one word	Address matches with a Character Match Percentage of 65 or more, at least one word matches in the first address line.
Country	The country name matches exactly.
No address	The address matches when it is missing in one or both of the records.
Address conflict	The addresses do not match at all. By default, this rule is only active for the first few primary identifier groups involving an exact name match. For example, if the addresses are different you must be confident that the names are the same and understand that it is a very loose match.

It is also possible to perform matching or elimination of Entity records using custom unique identifiers, see Section 5, "ID Matching."

5 ID Matching

The ID Matching rules in EDQ-CDS allow matching (or elimination) based solely on custom unique identifiers, without the need for a name match of some kind.

Matching and elimination is provided for Entity and Individual Matching, but not Address Matching.

Note:

- Unique ID (UID) matching is always performed before EID matching. Therefore, if two records are matched by unique identifiers, they cannot then be eliminated.
- These identifiers are always compared in standardized form; for example, values that differ only in case or additional non-alphanumeric character are considered identical. for example, the following values are identical for the purposes of ID matching:
  - AB123456789
  - ab123-456-789
  - ab12345 6789
  - ab#123456789

5.1 Unique ID Matching

The UID Match rules are held in the `[I005] UID` and `[E005] UID` match group of the Individual and Entity Match processes respectively. For example, for the match groups for Individual matches are as follows:

- [I005A] Match UID1
- [I005B] Match UID2
- [I005C] Match UID3

To use these rules, map the required data in the records to one or more of the **uid** attributes. The matching rules will always match two records sharing a common unique identifier, even if none of the other attributes match.

Note:

- The uid attributes accept multiple values in the form of a pipe delimited list. A match will be returned between two records if any one of a multiple set of attribute values is matched.
- Matching between uid attributes is not possible, for example, uid1 values cannot be matched with uid2 or uid3 values.

Example

The Passport Number field in a series of records is configured as the uid1 attribute. Therefore, the following records are returned as a match:

Record ID	First Name	Last Name	uid1 (Passport Number)	Match?
1	Fred	Smith	12345678	Yes
2	John	Doe	12345678	Yes

The following records with multiple values in the uid1 field are also matched:

Record ID	First Name	Last Name	uid1 (Passport Number)	Match?
1	Fred	Smith	12312312 | 67867867	Yes
2	John	Doe	67867867 | 23423423	Yes

The SSN field for the same set of records is configured as the uid2 attribute. The uid1 and uid2 fields are not cross matched; even though the uid1 value of Record 1 matches the uid2 value of Record 2:

Record ID	First Name	Last Name	uid1 (Passport Number)	uid2 (SSN)	Match?
1	Fred	Smith	12312312	67867867	No
2	John	Doe	67867867	12312312	No

5.2 Elimination ID Matching

The Elimination ID (EID) Match rules are held in the [ELIM015] EID ELIMINATIONS group of the Entity and Individual Match processes:

- [ELIM015A] ELIMINATE EID1
- [ELIM015B] ELIMINATE EID2

- [ELIM015C] ELIMINATE EID3

To use these rules, map the required data in the records to one or more of the eid attributes. The EID matching rules will always return a "No Match" result for two records that do not share a common value in an eid attribute, even if all other attributes match. The exception to this is if the two records are matched using a uid attribute, as UID matching is performed before EID matching.

Note:

- eid attributes accept multiple values in the form of a pipe delimited list. A "No Match" result will be returned between two records if none the values in an attribute are matched.
- Eliminating possible matches by comparing values between different eid attributes is not possible, for example, eid1 values cannot be compared with eid2 or eid3 values.

Example

The SSN field in a series of records is configured as the eid1 attribute. Therefore, the following records are eliminated as a possible match:

Record ID	First Name	Last Name	eid1 (SSN)	Eliminate?
1	John	Doe	12345678	Yes
2	John	Doe	87654321	Yes

The following records with multiple values in the eid1 field are also eliminated as a possible match, as none of the values match:

Record ID	First Name	Last Name	eid1 (SSN)	Eliminate?
1	John	Doe	12312312 | 23423423	Yes
2	John	Doe	45645645 | 67867867	Yes

The Passport field for the same set of records is configured as the eid2 attribute. The eid1 and eid2 fields are not compared, and therefore a "No Match" result is returned and the records are eliminated as a possible match:

Record ID	First Name	Last Name	eid1 (SSN)	eid2 (Passport Number)	Eliminate?
1	John	Doe	12312312	67867867	Yes
2	John	Doe	67867867	12312312	Yes

Finally, there are two identical values in the eid1 fields of the following records, and therefore they are *not* eliminated as a possible match:

Record ID	First Name	Last Name	eid1 (SSN)	Eliminate?
1	John	Doe	12312312 | 23423423	No
2	John	Doe	45645645 | 12312312	No

6 Address Matching

The rules for matching addresses include the use of pre-matching transformations and various matching comparisons in order to handle variance between different representations of what may be the same address, for example:

- Addresses containing abbreviated terms or suffixes.
- Character order and spelling differences/errors in addresses.
- Addresses with different levels of completeness.
- Addresses where extracted premise and subpremise match, and other components of the address are in a different order or missing on one side.

The following table lists all of the rules provided:

Address Match Rule Code	Address Match Rule Description
[A010]	Address exact, postal code exact
[A020]	Address exact, no postal code
[A030]	Address lines 1 and 2 exact, city exact, postal code exact
[A040]	Address lines 1 and 2 exact, city exact, postal code starts with
[A050]	Address all words, subpremise exact, premise exact, postal code exact
[A060]	Address all words, subpremise exact, premise exact, postal code no conflict
[A070]	Address 1 exact, address 2 no conflict, subpremise exact, premise exact postal code exact
[A080]	Address 1 exact, address 2 no conflict, subpremise exact, premise exact, postal code starts with
[A090]	Address 1 exact, address 2 no conflict, subpremise exact, premise exact, postal code no conflict
[A100]	Address all words typos, subpremise exact, premise exact, postal code exact
[A110]	Address all words typos, subpremise exact, premise exact, postal code no conflict
[A120]	Address 1 exact, address 2 no conflict, postal code exact
[A130]	Address 1 exact, address 2 no conflict, postal code starts with
[A140]	Address 1 exact, subpremise exact, premise exact, postal code exact
[A150]	Address 1 exact, subpremise exact, premise exact, postal code starts with
[A160]	Address 1 exact, subpremise no conflict, premise no conflict, postal code exact
[A170]	Address 1 exact, subpremise no conflict, premise no conflict, postal code starts with
[A180]	Address all words, subpremise no conflict, premise no conflict, postal code exact

Address Match Rule Code	Address Match Rule Description
[A190]	Address all words, subpremise no conflict, premise no conflict, postal code no conflict
[A200]	Address 1 all words, subpremise exact, premise exact, postal code exact
[A210]	Address 1 all words, subpremise exact, premise exact, postal code starts with
[A220]	Address 1 all words, subpremise no conflict, premise no conflict, postal code exact
[A230]	Address 1 all words, subpremise no conflict, premise no conflict, postal code starts with
[A240]	Address1 common string 7+, subpremise exact, premise exact, postal code exact
[A250]	Address all words, postal code exact
[A260]	Address similar, subpremise exact, premise exact, postal code exact
[A270]	Address 1 all words, address 2 no conflict, postal code exact
[A280]	Address 1 all words, address 2 no conflict, postal code starts with
[A290]	Address all words typos, postal code exact
[A300]	Address 1 exact, subpremise exact, premise exact, postal code no conflict
[A310]	Address 1 all words, subpremise exact, premise exact, postal code no conflict
[A320]	Address 1 exact, postal code exact
[A330]	Address 1 exact, postal code starts with
[A340]	Subpremise exact, premise exact; postal code exact
[A350]	Subpremise exact, premise exact, postal code starts with
[A360]	Address all words
[A370]	Address all words typos
[A380]	Address similar; postal code
[A390]	Address similar; first address one word

The following table provides examples of matches by Match Rule Code only, with the key fields highlighted in bold text where required:

Address Match Rule Code	Address Component	Record	Matched Record
[A010]	address1	901 GOLF CLUB RD	901 GOLF CLUB RD
	city	WESTWOOD	WESTWOOD
	subadminarea	PLUMAS	PLUMAS
	adminarea	CA	CA
	postalcode	96137	96137
	country	US	US
[A020]	As for [A010], but the **postalcode** field in both records is blank.		
[A030]	address1	1201 BEECH ST	1201 BEECH ST

Address Match Rule Code	Address Component	Record	Matched Record
	address2	APT 104F	APT 104F
	city	PALO ALTO	PALO ALTO
	subadminarea	**SANTA CLARA**	**SAN MATEO**
	adminarea	CA	CA
	postalcode	94303	94303
	country	US	US
[A040]	As [A030], except the v field in one address starts with the same characters as the other, but is not identical.		
[A050]	address1	**5 Hogskoleringen**	**Hogskoleringen 5**
	city	Trondheim	Trondheim
	adminarea		SØR-TRØNDELAG
	postalcode	**7491**	**7491**
	country	Norway	Norway
[A060]	As [A050], except one or both of the **postalcode** fields are blank.		
[A070]	address1	**Heinrichboeckingstr 10-14**	**Heinrichboeckingstr 10-14**
	address2	Service Zentrum Merzig	
	city	Saarbrücken	Saarbrücken
	adminarea		SAARLAND
	postalcode	66121	66121
	country	Germany	Germany
[A080]	Same as [A070], except the postalcode field in one address starts with the same characters as the **postalcode** field in the other, but is not identical.		
[A090]	Same as [A070], except one or both of the `postalcode` fields are blank.		
[A100]	address1	**HOGSKOLERINGE 5**	**HOGSKOLERINGEN 5**
	city	Trondheim	Trondheim
	postalcode	9491	9491
	country	Norway	Norway
[A110]	Same as [A100], except one or both of the **postalcode** fields are blank.		
[A120]	address1	**Marshfield Bank**	**Marshfield Bank**
	address2	**WOOLSTANWOOD**	
	city	Crewe	Crewe
	postalcode	**CW28UY**	**CW28UY**
	country	UK	UK
[A130]	Same as [A120], except the **postalcode** field in one address starts with the same characters as the **postalcode** field in the other, but is not identical.		
[A140]	address1	**Apt Y302**	**APT Y302**

Address Match Rule Code	Address Component	Record	Matched Record
	address2	**1605** Sherringtowne Ave	**1605** Sherington Ave
	city	NEWPORT BEACH	NEWPORT BEACH
	adminarea	Orange	Orange
	postalcode	**92663-9087**	**92663-9087**
	country	US	US
[A150]	Same as [A140], except the **postalcode** field in one address starts with the same characters as the **postalcode** field in the other, but is not identical.		
[A160]	address1	1728 **Corporate Xing**	1728 **Corporate Xing**
	address2	**Suite1**	
	city	O Fallon	O Fallon
	adminarea	ILLINOIS	IL
	postalcode	62269-3734	62269-3734
	city	US	US
[A170]	Same as [A160], except the **postalcode** field in one address starts with the same characters as the **postalcode** field in the other, but is not identical.		
[A180]	address1	Block 16	16 Dunsinane Ave
	address2	Dunsinane Avenue	
	address3	Dunsinane Industrial Estate	
	city	Dunsinane	Dunsinane
	postalcode	DD23QT	DD23QT
	country	UK	UK
[A190]	As [A180], except one or both of the **postalcode** fields are blank.		
[A200]	address1	**26701 QUAIL CRK**	**26701 QUAIL CRK APT 107**
	address2	**APT 107**	
	city	ALISO VIEJO	LAGUNA HILLS
	postalcode	92656-1089	92656-1089
	country	US	US
[A210]	Same as [A200], except the **postalcode** field in one address starts with the same characters as the **postalcode** field in the other, but is not identical.		
[A220]	address1	**Folkes Road**	**Unit 12 Folkes Road**
	address2	Hayes Trading Estate	Lye
	address3	Lye	
	city	Stourbridge	Stourbridge
	postalcode	**DY98RN**	**DY98RN**
	country	UK	UK
[A230]	Same as [A220], except the **postalcode** field in one address starts with the same characters as the **postalcode** field in the other, but is not identical.		

Address Match Rule Code	Address Component	Record	Matched Record
[A240]	address1	**101/61 NAWANAKORN INDUSTRY**	**101/61 NAVANAKORN INDUSTRY**
	address2	SELFLEMENT PHAHONYOTHIN	PAHOLYOTHIN KLONGNUENG
	city	KLONGLAUNG	KHLONG LUANG
	postalcode	**12120**	**12120**
	country	Thailand	Thailand
[A250]	address1	Blyth House	Blyth House
	address2	**130** Hordern Road	Hordern Road
	city	Wolverhampton	Wolverhampton
	postalcode	WV60HS	WV60HS
	country	UK	UK
[A260]	address1	**21001** State Route 739	**21001** Sr Rt **739**
	address2	7	
	city	Raymond	Raymond
	postalcode	**43067**	**43067**
	country	United States	United States
[A270]	address1	Lancaster House Aviation Way	Aviation Way
	address2		Southend Airport
	city	SOUTHEND ON SEA	SOUTHEND ON SEA
	postalcode	SS26UN	SS26UN
	country	UK	UK
[A280]	Same as [A270], except the **postalcode** field in one address starts with the same characters as the **postalcode** field in the other, but is not identical.		
[A290]	address1	**Blythe House**	**Blyth House**
	address2	130 **Hordern Road**	**Hordern Road**
	city	Wolverhampton	Wolverhampton
	postalcode	**WV60HS**	**WV60HS**
	country	UK	UK
[A300]	Same as [A140], except one or both of the **postalcode** fields are blank.		
[A310]	Same as [A200], except one of both of the **postalcode** fields are blank.		
[A320]	address1	**Network House**	**Network House**
	address2	1 Ariel Way	Wood Lane
	city	London	London
	postalcode	**W127SL**	**W127SL**
	country	UK	UK

Address Match Rule Code	Address Component	Record	Matched Record
[A330]	Same as [A320], except the **postalcode** field in one address starts with the same characters as the **postalcode** field in the other, but is not identical.		
[A340]	address1	College Business Park	College Business Park
	address2	Park	Coldhams Lane
	city	Cambridge	
	postalcode	**CB13HD**	**CB13HD**
	country	United Kingdom	United Kingdom
[A350]	Same as [A340], except the **postalcode** field in one address starts with the same characters as the **postalcode** field in the other, but is not identical.		
[A360]	address1	938 Miller St	Medical Ctr Blvd
	address2	**Medical Center Boulevard**	
	city	Winston Salem	Winston- Salem
	postalcode	27157	27157
	country	United States	United States
[A370]	address1	**Humberstone Avenue**	24 **Humberston Avenue**
	address2	**Humberstone**	**Humberston**
	city	GRIMSBY	GRIMSBY
	postalcode	DN364SX	DN364SP
	country	UK	UK
[A380]	address1	5**Sidings Court**	Greyfriars House
	address2	White Rose Way	**Sidings Court**
	city	DONCASTER	DONCASTER
	postalcode	**DN45NU**	**DN45NU**
	country	UK	UK
[A390]	address1	120 **Howard** St	120 **Howard** St
	address2		STE 200
	city	San Fransisco	San Fransisco
	adminarea	CA	CA
	postalcode	94105-1622	94105-1615
	country	United States	United States

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.

Customer Data Services Pack Matching Guide, Release 11g R1 (11.1.1.7)
E40737-01