# Oracle® Financial Services

# Trade-Based Anti Money Laundering Matching Guide

Release 8.1.2.11.0
G52583-01
February 2026

Oracle Financial Services Trade-Based Anti Money Laundering Matching Guide, Release 8.1.2.11.0

G52583-01

# Contents

# Document Control

This topic lists the changes that have been made to this guide in each release.

**Table Revision History**

| Date | Edition | Description |
|---|---|---|
| February 2026 | First edition of 8.1.2.11.0 | There are no content changes to this guide in this release. The look and feel of the document has been updated. |
| August 2025 | First edition of 8.1.2.10.0 | There are no content changes to this guide in this release. The look and feel of the document has been updated. |
| February 2025 | First edition of 8.1.2.9.0 | There are no content changes to this guide in this release. The look and feel of the document has been updated. |
| August 2024 | First edition of 8.1.2.8.0 | There are no changes to this guide in this release. |
| February 2024 | First edition of 8.1.2.7.0 | There are no changes to this guide in this release. |
| December 2019 | First edition of 8.0.8.0.0 | There are no changes to this guide in this release. |
| December 2018 | First edition of 8.0.7.0.0 | This is the first edition of this guide. |

1

# About This Guide

This guide provides information on webservices matching, individual matching, vessel matching, aircraft matching, and entity matching.

## Audience

The Trade-Based Anti Money Laundering Matching Guide is designed for the following users:

- **Analyst:** This user works on the transactions within the application frequently. This user’s specific role determines what they can view and perform within the application.
- **Supervisor:** This user works on the transactions within the application on a daily basis and is typically a higher level Analyst or Compliance Officer.
- **Admin:** This user performs the activities related to administration.
- **Case Analyst:** This user works on the cases within the application frequently. This user’s specific role (that is, Case Analyst I or Case Analyst II) determines what this user can see and do within the application.
- **Case Supervisor:** This user works on cases within the application on a daily basis and is typically a higher level Analyst or Compliance Officer.
- **Case Executive:** This user may not be involved in the day-to-day analysis of cases; however, this user can view many areas within the application and can perform only a limited set of actions.
- **Case Auditor:** This user has broad viewing rights for cases within the application; however, this user can perform a limited set of actions based on the user’s role (that is, Case Internal Auditor or Case External Auditor).
- **Case Viewer:** This user has specific viewing rights to the cases. This user’s specific role (that is, Case Viewer I or Case Viewer II) determines what this user can see within the application.

## 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 that have purchased support 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.

## Related Resources

See these Oracle resources:

- Oracle Financial Services Behavior Detection Installation Guide
- Oracle Financial Services Trade-Based Anti Money Laundering Administration Guide

## Conventions

The following text conventions are used in this document.

| Convention | Meaning |
|---|---|
| **boldface** | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary. |
| *italic* | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values. |
| `monospace` | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

2

# About Oracle Financial Services Trade-Based Anti Money Laundering

This topic provides a brief overview of Oracle Financial Services Trade-Based Anti Money Laundering (TBAML) in terms of its architecture and operations.

### About TBAML

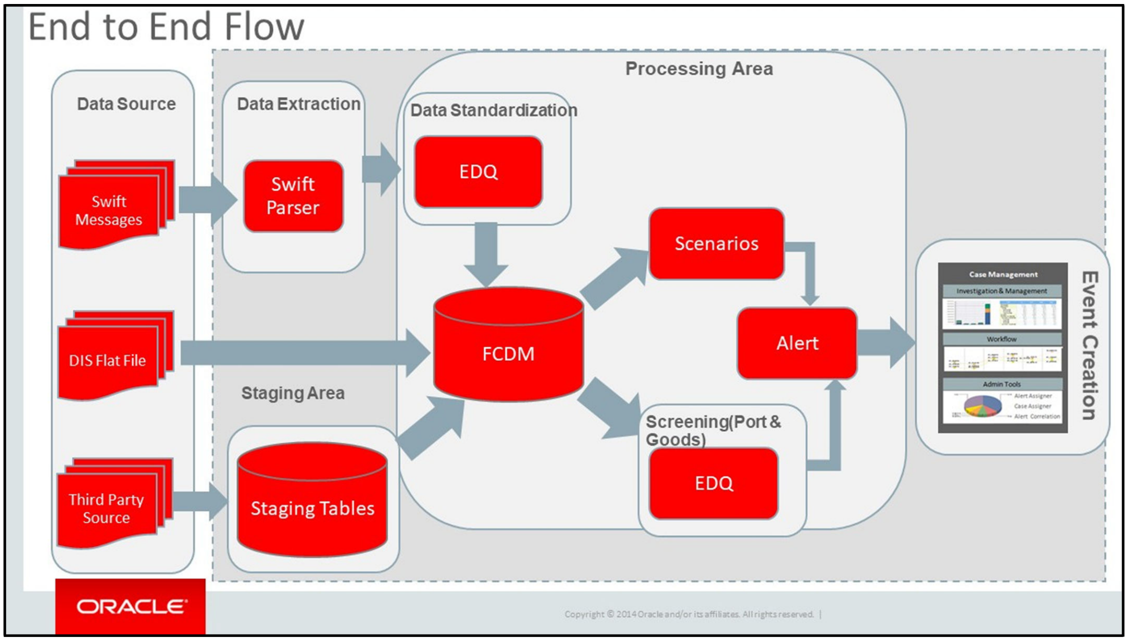
Oracle Financial Services Trade-Based Anti Money Laundering (TBAML) offers a comprehensive compliance solution to:

- Efficiently screen goods, ports and involved party names extracted from SWIFT MT messages as well as on federal trade data against various lists such as sanctions lists, watch lists, and so on.
- Continuously monitor trade finance transactions using a risk based approach for potential TBML activities, such as TBML red flag topologies, by assessing the trade finance customer, transactions (specifically goods, contract amount, goods price), and involved counterparties (name and address).

### TBAML Architecture

An architecture is a blueprint of all the parts that together define the system: its structure, interfaces, and communication mechanisms. A set of functional views can describe an architecture.

**Figure 2-1    TBAML Architecture**



TBAML extracts data from SWIFT messages via the Swift Parser functionality. That data, and data provided by the Oracle client via DIS File or another source, is fed into staging tables and then into the FCDM where the data is either standardized (Port) and screened (Port, Goods, Name and Address) through EDQ functionality, or run through scenarios to generate an FCM event.

**Deployment View**

The TBAML architecture from the perspective of its deployment illustrates deployment of the major subsystems across servers. Additionally, the deployment view shows the primary communications links and protocols between the processing nodes.

The complex interactions between the components of the Alert & Case Management tiers becomes apparent in the deployment view. The Alert & Case Management tiers require the following:

- Web browser
- Web server
- Web application server

Alert & Case Management tiers use OFSAAI for handling both authentication and authorization. The Alert & Case Management subsystem also supports the use of an External Authentication Management (EAM) tool to perform user authentication at the web server, if a customer requires it. TBAML components can operate when deployed on a single computer or when distributed across multiple computers. In addition to being horizontally scalable, TBAML is vertically scalable in that replication of each of the components can occur across multiple servers.

**Security View**

The security view describes the architecture and use of security features of the network in a TBAML architecture deployment. TBAML uses an inbuilt Security Management System (SMS) for its authentication and authorization. The SMS has a set of database tables which store information about user authentication.

Installation of 128-bit encryption support from Microsoft can secure the web browser. Oracle encourages using the Secure Socket Layer (SSL) between the web browser and web server for login transaction, while the web Application server uses a browser cookie to track a user's session. This cookie is temporary and resides only in browser memory. When the user closes the browser, the system deletes the cookie automatically.

TBAML uses Advanced Encryption Standard (AES) security to encrypt passwords that reside in database tables in the ATOMIC schema on the database server and also encrypts the passwords that reside in configuration files on the server.

The EAM tool is an optional third-party pluggable component of the security view. The tool’s integration boundaries provide an Authorization header, form field with principal, or embedded principal to the web Application server through a web server plug-in. The tool also passes the same user IDs that the TBAML directory server uses.

## 2.1 Operations

As the administrator, you coordinate the overall operations of TBAML: Data Management, Behavior Detection, and Post-Processing.

In a production environment, an Oracle client typically establishes a processing cycle to identify occurrences of behaviors of interest (that is, scenarios) at a specific frequency. Each cycle begins with Data Management, Behavior Detection, and Post-Processing, which prepares the detection results for presentation for the users.

Several factors determine specific scheduling of these processing cycles, including availability of data and the nature of the behavior that the system is to detect. The following sections describe each of the major steps in a typical production processing cycle:

1. Start Batch
2. Managing Data
3. Behavior Detection
4. Post-Processing
5. End Batch

### Start Batch

Using the Batch Control Utility, you can manage the beginning of the batch process.

### Managing Data

The Ingestion Manager controls the Data Management process. The Data Interface Specification (DIS) contains specific definition of the types and format of business data that can be accepted for ingestion.

The Ingestion Manager supports files and messages for the ingestion of data. Data Management involves receiving source data from an external data source in one of these forms. The Ingestion Manager validates this data against the DIS, applies required derivations and aggregations, and populates the database with the results.

### Behavior Detection

During Behavior Detection, OFSBD Algorithms control the scenario detection process. The Detection Algorithms search for events and behaviors of interest in the ingested data in the FCDM. Upon identification of an event or behavior of interest, the algorithms record a match in the database.

A match is created by executing scenarios. These scenarios are used to detect the behaviors of interest that correspond to patterns or the occurrences of pre-specified conditions in business data. The process also records additional data that the analysis of each match may require.

### Post-Processing

During post-processing of detection results, Behavior Detection prepares the detection results for presentation to users. Preparation of the results depends upon the following processes:

- Match Scoring: Computes a ranking for scenario matches indicating a degree of risk associated with the detected event or behavior.
- Alert Creation: Packages the scenario matches as units of work (that is, events), potentially grouping similar matches together, for disposition by end users. This is applicable when multiple matches with distinct scores are grouped into a single event.
- Alert Scoring: Ranks the events (including each match within the events) to indicate the degree of risk associated with the detected event or behavior.
- Highlight Generation: Generates highlights for events that appear in the event list in the behavior detection subsystem and stores them in the database.

- Historical Data Copy: Identifies the records against which the current batch's scenario runs generated events and copies them to archive tables. This allows for the display of a snapshot of information as of the time the event behavior was detected.
- Alert Correlation: Uncovers relationships among events by correlating events to business entities and subsequently correlating events to each other based on these business entities. The relationships are discovered based on configurable correlation rule sets.

**End Batch**

The system ends batch processing when processing of data from the Oracle client is complete. The Alert & Case Management subsystem then controls the event and case management processes. See the Behavior Detection User Guide and Enterprise Case Management User Guide for more information.

## 2.2 Utilities

TBAML database utilities enable you to configure and perform pre-processing and post-processing activities.

**Batch Utilities**

Behavior Detection database utilities enable you to configure and perform batch-related system pre-processing and post-processing activities.

- Alert Purge Utility: Provides the capability to remove erroneously generated matches, events, and activities.
- Batch Control Utility: Manages the start and termination of a batch process (from Data Management to event post-processing) and enables access to the currently running batch.
- Calendar Manager Utility: Updates calendars in the system based on pre-defined business days, holidays, and days off, or non-business days.
- Data Retention Manager: Provides the capability to manage the processing of partitioned tables in Behavior Detection. This utility purges data from the system based on configurable retention period defined in database.
- Database Statistics Management: Manages Oracle database statistics. These statistics determine the appropriate execution path for each database query.
- Notification: Enables you to configure users to receive UI notifications based upon actions taken on events or cases to which they are associated or when the event or case is nearing a due date.
- Truncate Manager: Truncates tables that require complete replacement of their data.

**Administrative Utilities**

The following database utilities that configure and perform system pre-processing and post-processing activities are not tied to the batch process cycle:

- Scenario Migration Utility: Extracts scenarios, datasets, networks, and associated metadata from a database to flat files and loads them into another environment.
- Threshold Editor: Allows you to run the same scenario multiple times against a variety of sources (for example, exchanges, currencies, or jurisdictions) with separate threshold values for each source.

# 3 Introduction to Matching

Oracle Financial Services TBAML provides a flexible and customizable strategy for matching customer records to watch list records. Sanctions screening typically requires the business to employ tightly-defined, zero tolerance matching policies which will identify every possible match against a sanctions list.

Oracle Financial Services TBAML therefore employs a range of clustering strategies and matching rules. These can be enabled and disabled as needed, to tune the behavior of Oracle Financial Services TBAML to your requirements.

In general, the looser the match rule, the more likely it is to raise false positives. It is not possible to eliminate all false positives, especially if there is a requirement to identify all true matches. Tuning the matching strategy is therefore a trade-off between the proportion of true matches which are not detected and the work required to manually eliminate false positives. This will be evident in the examples in this document.

## 3.1 General Matching Strategy

This section provides a brief description of the general strategy used in Oracle Financial Services TBAML. It consists of three main components: identifier preparation, clustering and matching.

### Identifier Preparation

There are some differences between the structure of data sets that always need to be normalized before clustering and matching, so that the matching process does not need to repeat the configuration of transformations on each comparison.

Identifier preparation is used to ensure that the records conform to a pre-defined data structure which can be used by the rest of the matching process, and also to eliminate common forms of variance between the records (such as spelling variants of given names and abbreviations of frequently-used tokens).

### Clustering

Clustering is used to minimize the work that must be performed by the final stage of matching. It works by splitting the working and reference data into wide 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 later in the matching process.

If very wide 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 power is required to compare all the clustered records by brute force. A tighter clustering strategy will result in smaller clusters, with fewer records per cluster. This results in reduced processing requirements for row-by-row comparisons, but increases the likelihood that some true matches will not be detected.

### Matching

Once the working and watch list records have been divided into clusters, the rows within each cluster are compared to one another according to the match rules defined for the matching processor. Each match rule defines a set of criteria, specified as comparisons, that the pair of records must satisfy in order to qualify as a match under that rule. The rules are applied as a decision table, so if a pair of records qualifies as a match under a rule higher in the table, it will not be compared using any rules below that. All rules are configured to operate on a case-insensitive basis. Unless stated otherwise, all noise and whitespace characters are removed or normalized before matching.

## 3.2 Configuring TBAML for Different Scenarios

As previously mentioned, Oracle Financial Services TBAML includes clusters and matching rules that are suited to various screening requirements.

Tuning TBAML to match your policies should be undertaken carefully and under the supervision of a risk and compliance expert, with knowledge of your business requirements and the relevant legislation.

The following general points may be useful when tuning the behavior of TBAML:

- Some organizations use a zero-tolerance policy for individual name matches. Such a policy typically requires that all potential name matches must be manually reviewed, irrespective of the rest of the data associated with the record. If such a policy is in place, you must enable the conflict rules in individual matching.
- In general, using a loose clustering strategy will result in relatively few clusters, each containing many records. This reduces the potential for missing true matches, and increases the chance of false positives. It is also expensive in terms of processing requirements, as every record in the cluster must be directly compared with every other record in that cluster.
- Conversely, a tight clustering strategy will result in a relatively large number of clusters, each containing fewer records. This increases the potential for missing true matches, and decreases the chance of false positives. It also reduces the overall cost of the processing requirements, as relatively few direct comparisons are required.

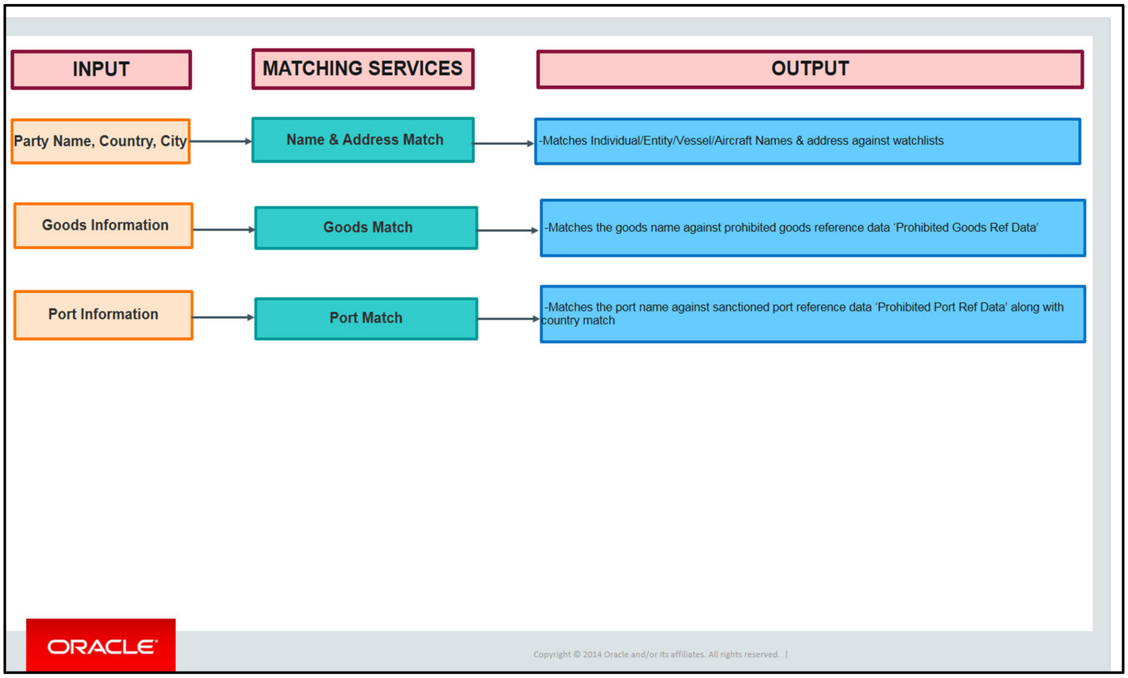
# 4

# Matching Webservices

This section gives an overview of Matching Webservices for Trade-Based Anti Money Laundering.

The following diagram depicts the input and output for these Webservices.

**Figure 4-1 Matching Webservices**



INPUT

MATCHING SERVICES

OUTPUT

Party Name, Country, City → Name & Address Match → -Matches Individual/Entity/Vessel/Aircraft Names & address against watchlists

Goods Information → Goods Match → -Matches the goods name against prohibited goods reference data 'Prohibited Goods Ref Data'

Port Information → Port Match → -Matches the port name against sanctioned port reference data 'Prohibited Port Ref Data' along with country match

## Name and Address Matching

This matching webservice is used for the party name and address matching, such as orderer and beneficiary. Party can be an individual, entity, vessel, or aircraft. The webservice takes the party name, country (if available in the message) and city (if available in the message) as an input and matches them against the watchlist records.

For information on the reference data for watchlists, see the Oracle Financial Services Trade-Based Anti Money Laundering Administration Guide.

## Trade Goods Matching

This matching webservice is used to extract the trade goods name and match it name against the prohibited goods list. This list provides country-wise data.

For information on the reference data for watchlists, see the Oracle Financial Services Trade-Based Anti Money Laundering Administration Guide.

### Trade Port Matching

This matching webservice is used to extract the trade port name and match it to the country while screening. This is done because two countries may not have the same port name, so this avoids duplication. The port name is also matched against the sanctioned port reference data.

For information on the reference data for watchlists, see the Oracle Financial Services Trade-Based Anti Money Laundering Administration Guide.

5

# Name and Address Matching

This section details the default configuration of Name and Address screening against sanctions lists or watch lists.

The matching strategy for entities in Oracle TBAML raises a possible match if there is an exact match or a fuzzy name match to a normal (non-acronym) entity name, or if there is an exact match to an acronym entity name. The fuzzy entity name matching algorithms include some of the following techniques:

- Standardizing entity names (for example, different forms of company name suffixes are standardized to a common form)
- Ignoring insignificant name tokens
- Typo tolerance
- Allowance for missing name tokens
- Allowance for different tokenization of the name

## 5.1 Identifier Preparation

The following identifiers are prepared for use in the individual and Entity matching process.

ⓘ **Note**

For Identifier preparation, Vessel and Aircraft are considered Entities.

**Table 5-1 Individual and Entity Identifier Preparation**

| Identifier Description | Standard Prepared Attribute Name | Summary of Preparation Logic |
|---|---|---|
| Individual Family Name | dnFamilyName | A normalized version of the family name (see the Name Normalization section). |
| Individual Full Name | dnFullName | A concatenation of the given names and family name, separated using spaces. |
| Original Script Name | dnOriginalScriptName | A whitespace normalized version of the original script name. |
| dnCity | dnCity | A pipe-separated list of cities associated with the individual data. |
| dnAddressCountr yCode | dnAddressCountryCode | A space separated list of standard 2- character country codes. |
| dnEntityName | dnEntityName | The original entity name, after Name Normalization. |

**Table 5-1 (Cont.) Individual and Entity Identifier Preparation**

| Identifier Description | Standard Prepared Attribute Name | Summary of Preparation Logic |
|---|---|---|
| Individual Given Names | dnGivenNames | A space-separated list of the first and middle names of the individual, after normalization (see the Name Normalization section). |

The following sections describe the data preparation strategy for each of these identifiers.

### Name Normalization

The individual, entity, vessel, and aircraft names are normalized using the following logic:

- Standardization of accented characters.
- Replacement of non-alpha (A-Z or a-z) characters with spaces.

> **Note**
>
> - If data is matched in the original language against original script names in the watch lists, then the appropriate character ranges must be removed from the Name Noise Characters Reference Data so that they are not replaced.
> - If transliteration of data is done before matching, then transliteration must also be done before name normalization.

- Normalization of whitespace.
- Conversion to upper case.

Note that the purpose of these transformations is not to create the most ‘correct’ name. For example, hyphens may be used in names in a number of ways, such as in a double-barreled surname, or as an alternative for a space when a surname has a qualifier (common in the World-Check data file).

In the former case, one might ideally want to preserve the hyphen, and in the latter case replace it with a space. In general, however, additional spaces in names will not cause names to mismatch, whereas different characters could. The following table provides some examples.

**Table 5-2 Name Normalization**

| Input Data | | Identifiers | | |
|---|---|---|---|---|
| **Forename** | **Surname** | **dnGivenNames** | **dnFamilyName** | **dnFullName** |
| Darwen | MANN`A | DARWEN | MANN A | DARWEN MANN A |
| Badr bin Saud bin Harib | AL- BUSAIDI | BADR BIN SAUD BIN HARIB | AL BUSAIDI | BADR BIN SAUD BIN HARIB AL BUSAIDI |
| A. Arnaldo G. | TAVEIRA | A ARNALDO G | TAVEIRA | A ARNALDO G TAVEIRA |
| Jose Mardônio | DA COSTA** | JOSE MARDONIO | DA COSTA | JOSE MARDONIO DA COSTA |

**Table 5-2 (Cont.) Name Normalization**

| Input Data | | Identifiers | | |
|---|---|---|---|---|
| **Forename** | **Surname** | **dnGivenNames** | **dnFamilyName** | **dnFullName** |
| Carmelo | Raschellà | CARMELO | RASCHELLA | CARMELO RASCHELLA |

### City and Country Identifiers

City and country values are derived from the source data wherever possible. There may be multiple possible cities or countries associated with an individual, perhaps because an individual resides in more than one country, has dual nationality, or resides in a different country from their nationality.

Country values are prepared as a space-separated list of two-character country codes in the `dnAllCountryCodes` attribute.

City values (which may contain spaces, for example, 'New York') are prepared as a pipe-separated list of cities in the `dnCity` attribute.

# 5.2 Clustering

Oracle Financial Services Trade-Based Anti Money Laundering provides clusters for matching individuals and entities to watch lists during Sanctions screening.

These clusters can be activated or deactivated, as required, and different cluster limits can be configured. The following Cluster Methods can be used:

- Individual Family Name
- Individual Full Name Metaphone
- Individual Given Names
- Individual Full Name Trim
- Individual Initials
- Original Script Name
- Entity Name Tokens
- Entity Name Meta
- Entity Name Trim
- Entity Start End Name Tokens

> **Note**
>
> This lists the default configuration of Real-Time screening processes, but these may be customized independently of one another.

The data used to create the clusters is created before matching by the preparation process. In all cases, the clusters use the prepared and normalized name attributes `dnGivenNames`, `dnFamilyName` , `dnFullName`, `dnEntityName`, and `dnOriginalScriptName`. For further information see Name Normalization.

## 5.2.1 Family Name Cluster (dnClusterFamilyName)

The Family Name cluster provides a backup to the full name clusters. This is especially important where the given name data is incomplete, making it difficult to form a complete cluster key for two names.

For example, the following three example records do not share any Full Name cluster keys, due to the initials in the second record and the spacing and spelling variations seen throughout:

**Table 5-3 Family Name Cluster**

| dnFullName | Name Tokens and Trimmed Values | Cluster Keys | dnClusterFullNameTrim |
|---|---|---|---|
| STEPHEN JEQE NKOMO | JEQE\| JEQ<br>NKOMO\| NKO<br>STEPHEN\| STE | JEQNKO JEQSTE NKOSTE | JEQNKO\|JEQSTE\|NKOSTE |
| S J NKOMO | S\| S<br>NKOMO\| NKO<br>J\| J | NKO | NKO |
| STEPHEN JEKE N KOMO | JEKE\| JEK<br>KOMO\| KOM<br>N\| N<br>STEPHEN\| STE | JEKKOM JEKSTE KOMSTE | JEKKOM\|JEKSTE\|KOMSTE |

Clustering only on the family name circumvents this issue, but results in large clusters and a concomitant increase in the processing required to cross-check all the records.

The **Family Name** cluster builder counters spacing and punctuation differences by generating Metaphone keys for all tokens of the family name, AND the whole of the family name after all white space is trimmed. This is to ensure that family names such as those in the last two records in the example table below are all clustered together despite the spacing differences.

The default logic of the cluster builder is as follows:

1. Trim all white space from the normalized family name.
2. Apply the **Metaphone**transformation to the result, outputting a key with a length of up to 4 characters.
3. Strip common name qualifiers from the normalized family name, such as Abd,Al.
4. Split the family name into several name tokens, using a space delimiter.

> **Note**
>
> Many other punctuation and noise characters are normalized to spaces before generating the cluster. For more information see Name Normalization.

5. Apply the Metaphone transformation to each name token, outputting a key with a length of up to 4 characters. If there were no tokens remaining after stripping common name qualifiers then apply the Metaphone transformation to the each name token of the original normalized family name.
6. Concatenate all the generated Metaphone keys.

7. Deduplicate the list of keys.

The following table provides some examples.

**Table 5-4 Metaphone Transformations for Family Name Cluster**

| dnFamilyName | Tokens Derived from dnFamilyName | Metaphone Transformations | dnClusterFamilyName |
|---|---|---|---|
| ZHONG | ZHONG | JNK | JNK |
| XIAOJIAN | XIAOJIAN | SJN | SJN |
| ABACHE | ABACHE | APX | APX |
| ABANDA | ABANDA | APNT | APNT |
| ABD AL HAFIZ | HAFIZ ABDALHAFIZ | HFS APTL | HFS\|APTL |
| AL BUTHE | BUTHE ALBUTHE | P0 ALP0 | P0\|ALP0 |
| AL | AL | AL | AL |
| SOLEIMAN HAMAD | SOLEIMAN HAMAD SOLEIMANHAMAD | SLMN HMT SLMN | SLMN\|HMT |
| GOODRIDGE | GOODRIDGE | KTRJ | KTRJ |
| GOODRICH SR | GOODRICH SR GOODRICHSR | KTRX SR KTRK | KTRX\|SR\|KTRK |
| NKOMO | NKOMO | NKM | NKM |
| N KOMO | N KOMO NKOMO | N KM NKM | N\|KM\|NKM |

## 5.2.2 Individual Full Name Metaphone Pairs Cluster (dnClusterFullNameMeta)

The Full Name Metaphone Pairs cluster uses the normalized full name for the individual to generate a cluster key for every pair of names within the full name.

The default logic of this is as follows:

1. Split the normalized full name into several name tokens, using space as a delimiter.

> ⓘ **Note**
>
> Many other punctuation and noise characters are normalized to spaces before generating the cluster. For further information see Name Normalization.

2. Sort the name tokens alphabetically.
3. Apply the Metaphone transformation (the standard double-metaphone algorithm) to each name token, outputting a key with a length of up to three characters.
4. Concatenate the Metaphone values, generating a final key value for each distinct pair of tokens.
5. Deduplicate the list of keys.

The following table provides some examples.

**Table 5-5 Full Name Metaphone Pairs Cluster**

| dnFullName | Name Tokens and Metaphone values | Distinct Cluster Keys | dnClusterFullNameMeta |
|---|---|---|---|
| XIAO JIAN ZHONG | JIAN \| JN<br>XIAO \| S<br>ZHONG\| JNK | JNS JNJNK SJNK | JNS\| JNJNK \|SJNK |
| ZHONG XIAOJIAN | XIAOJIAN \| SJN<br>ZHONG \| JNK | SJNJNK | SJNJNK |
| MOHAMMED SANI ABACHE | ABACHE \| ABX<br>MOHAMMED \| MHM T<br>SANI \| SN | APXMHM APXSN MHMSN | APXMHM\| APXSN\| MHMSN |
| JOSEPH TSANGA ABANDA | ABANDA \| APNT<br>JOSEPH \| JSF<br>TSANGA \| TSNK | APNJSF APNTSN JSFTSN | APNJSF\| APNTSN\| JSFTSN |
| ABD AL WAHAB ABD AL HAFIZ | ABD \| APT<br>ABD \| APT<br>AL \| AL<br>AL \| AL<br>HAFIZ \| HFS<br>WAHAB \| AHP | APTAPT APTAL APTHFS APTAHP ALAL ALHFS ALAHP HFSAHP | APTAPT\| APTAL\| APTHFS \| APTAHP\| ALAL\| ALHFS \| ALAHP\| HFSAHP |
| SULIMAN HAMD SULEIMAN AL BUTHE | AL \| AL<br>BUTHE \| P0<br>HAMAD \| HMT<br>SULEIMAN \| SLMN<br>SULIMAN \| SLMN | ALP0 ALHMT ALSLM P0HMT P0SLM HMTSLM SLMSLM | ALP0\| ALHMT\| ALSLM\| P0HMT\| P0SLM\| HMTSLM \| SLMSLM |
| AL BUTHE SOLEIMAN HAMAD | AL \| AL<br>BUTHE \| P0<br>HAMAD \| HMT<br>SOLEIMAN \| SLMN | ALP0 ALHMT ALSLM P0HMT P0SLM HMTSLM | ALP0\| ALHMT\| ALSLM\| P0HMT\| P0SLM\| HMTSLM |
| REGINALD B GOODRIDGE | B \| P<br>GOODRIDGE \| KTRJ<br>REGINALD \| RJNLT | KTRRJN<br>**NOTE:** Initials are ignored by default when generating cluster key | KTRRJN |
| REGINALD B SR GOODRICH | B \| P<br>GOODRIDGE \| KTRJ<br>REGINALD \| RJNLT<br>SR \| SR | KTRRJN KTRSR RJNSR<br>**NOTE:** Initials are ignored by default when generating cluster keys | KTRRJN\| KTRSR\| RJNSR |
| STEPHEN JEQE NKOMO | JEQE \| JK<br>NKOMO \| NKM<br>STEPHEN \| STFN | JKNKM JKSTF NKMSTF | JKNKM\| JKSTF\| NKMSTF |
| S J NKOMO | J \| J<br>NKOMO \| NKM<br>S \| S | NKM<br>Initials are ignored by default when generating cluster keys | NKM |
| STEPHEN JEKE N KOMO | JEKE \| JK<br>KOMO \| KM<br>N \|N<br>STEPHEN \| STFN | JKKM JKSTF KMSTF | JKKM\| JKSTF\| KMSTF |

## 5.2.3 Individual Given Names Cluster (dnClusterGivenNames)

The Given Names cluster provides a further backup to the remaining clusters, especially to deal with cases where names are not necessarily well-structured into family and given names.

> **Note**
>
> Depending on the quality and culture of the name information, this cluster will often not be required. You can test the number of additional alerts identified by the cluster by running matching with this cluster disabled, and then running with it enabled. Comparing the new relationships against the old will highlight the relationships identified by using this cluster.

The default logic of the cluster builder is as follows:

1. Split the normalized full name into several name tokens, using space as a delimiter.

   > **Note**
   >
   > Many other punctuation and noise characters are normalized to spaces before generating the cluster. For more information see Name Normalization.

2. Standardize the normalized given names before clustering. This ensures, for example, that names such as 'William' and 'Bill' will be clustered together, although their raw Metaphone values are not the same. A space delimiter is used to split the name before standardizing.
3. Apply the Metaphone transformation to the whole of the given names value after token standardization, outputting a key with a length of up to 4 characters.

The following table provides some examples.

**Table 5-6 Given Names Cluster**

| dnGivenNames | Metaphone Values | dnClusterGivenNames |
|---|---|---|
| XIAO JIAN | SJN | SJN |
| ZHONG | JNK | JNK |
| MOHAMMED SANI | MHMT | MHMT |
| JOSEPH TSANGA | JSFT | JSFT |
| ABD AL WAHAB | APTL | APTL |
| SULIMAN HAMD SULEIMAN | SLMN | SLMN |
| AL BUTHE | ALP0 | ALP0 |
| REGINALD B | RJNL | RJNL |
| STEPHEN JEQE | STFN | STFN |
| S J | SJ | SJ |
| STEPHEN JEKE | STFN | STFN |

## 5.2.4 Individual Full Name Trim Pairs Cluster (dnClusterFullNameTrim)

On occasion, two names which are close matches may not generate a common cluster key using the Full Name Metaphone Pairs cluster.

Consider the following example records:

**Table 5-7 Full Name Trim Pairs Cluster**

| dnFullName | Name Tokens and Metaphone Values | Distinct Cluster Keys | dnClusterFullNameMeta |
|---|---|---|---|
| XIAO JIAN ZHONG | JIAN \| JN<br>XIAO \| S<br>ZHONG \| JNK | JNS JNJNK SJNK | JNS \| JNJNK \| SJNK |
| ZHONG XIAOJIAN | XIAOJIAN \| SJN<br>ZHONG \| JNK | SJNJNK | SJNJNK |

These two records are a possible name match. However, the Full Name Metaphone Pairs cluster does not produce a common cluster key for the pair because the tokens ‘Xiao’ and ‘Xiaojian’ yield different three character Metaphone keys.

In order to match these cases efficiently, a Full Name Trim Pairs cluster is prepared in a similar way to the primary cluster, but without applying a Metaphone transformation. This allows for typos and spacing differences in the names, but is ‘left-biased’; that is, it demands that the first few characters of the names match.

The logic of the cluster is as follows:

1. Split the normalized full name into name tokens, using space as a delimiter.
2. Sort the name tokens alphabetically.
3. Apply the Trim Characters transformation to each name token, outputting a key with a length of (up to) 3 characters.
4. Concatenate the trimmed values, generating a final key value for each distinct pair of tokens.
5. Deduplicate the list of keys.

The following table provides some examples.

**Table 5-8 Trim Characters for Full Name Trim Pairs Cluster**

| dnFullName | Name Tokens and Trimmed Values | Cluster Keys | dnClusterFullNameTrim |
|---|---|---|---|
| XIAO JIAN ZHONG | JIAN \| JIA<br>XIAO \| XIA<br>ZHONG \| ZHO | JIAXIA JIAZHO XIAZHO | JIAXIA \| JIAZHO \| XIAZHO |
| ZHONG XIAOJIAN | XIAOJIAN \| XIA<br>ZHONG \| ZHO | XIAZHO | XIAZHO |
| MOHAMMED SANI ABACHE | ABACHE \| ABA<br>MOHAMMED \| MOH<br>SANI \| SAN | ABAMOH ABASAN MOHSAN | ABAMOH \| ABASAN \| MOHSAN |
| JOSEPH TSANGA ABANDA | ABANDA \| ABA<br>JOSEPH \| JOS<br>TSANGA \| TSA | ABAJOS ABATSA JOSTSA | ABAJOS \| ABATSA \| JOSTSA |

**Table 5-8 (Cont.) Trim Characters for Full Name Trim Pairs Cluster**

| dnFullName | Name Tokens and Trimmed Values | Cluster Keys | dnClusterFullNameTrim |
|---|---|---|---|
| ABD AL WAHAB ABD AL HAFIZ | ABD \| ABD<br>ABD \| ABD<br>AL \| AL<br>AL \| AL<br>HAFIZ \| HAF<br>WAHAB \| WAH | ABDABD ABDAL ABDHAF ABDWAH ALAL ALHAF ALWAH HAFWAH | ABDABD \| ABDAL \| ABDHAF \| ABDWAH \| ALAL \| ALHAF \| ALWAH \| HAFWAH |
| SULIMAN HAMD SULEIMAN AL BUTHE | AL \| AL<br>BUTHE \| BUT<br>HAMD \| HAM<br>SULEIMAN \| SUL<br>SULIMAN \| SUL | ALBUT ALHAM ALSUL ALSUL BUTHAM BUTSUL HAMSUL SULSUL | ALBUT \| ALHAM \| ALSUL \| BUTHAM \| BUTSUL \| HAMSUL \| SULSUL |
| AL BUTHE SOLEIMAN HAMAD | AL \| AL<br>BUTHE \| BUT<br>HAMD \| HAM<br>SOLEIMAN | ALBUT ALHAM ALSOL BUTHAM BUTSOL HAMSOL | ALBUT \| ALHAM \| ALSOL \| BUTHAM \| BUTSOL \| HAMSOL |
| REGINALD B GOODRIDGE | B \| B<br>GOODRIDGE \| GOO<br>REGINALD \| REG | GOOREG | GOOREG |
| REGINALD B SR GOODRICH | B \| B<br>GOODRICH \| GOO<br>REGINALD \| REG<br>SR \| SR | GOOREG GOOSR REGSR<br>**NOTE:** Initials are ignored by default when generating cluster keys. | GOOREG \| GOOSR \| REGSR |
| STEPHEN JEQE NKOMO | JEQE \| JEQ<br>NKOMO \| NKO<br>STEPHEN \| STE | JEQNKO JEQSTE NKOSTE | JEQNKO \| JEQSTE \| NKOSTE |
| S J NKOMO | S \| S<br>NKOMO \| NKO<br>J \| J | NKO<br>**NOTE:** Initials are ignored by default when generating cluster keys. | NKO |
| STEPHEN JEKE N KOMO | JEKE \| JEK<br>KOMO \| KOM<br>N \| N<br>STEPHE \| STE | JEKKOM JEKSTE KOMSTE<br>**NOTE:** Initials are ignored by default when generating cluster keys. | JEKKOM \| JEKSTE \| KOMSTE |

## 5.2.5 Individual Initials (dnClusterInitials)

The **First Initial Last Name** cluster provides a clustering method to group together names that share the same first name initial and last name, and allows some variation for transposed names.

The default logic of the cluster builder is as follows:

1. Split the normalized given names into several name tokens, using a space character as the delimiter.

2. Split the normalized family name into several name tokens, using a space character as the delimiter.
3. Generate the cluster key value as follows:
   - If there are two or more characters in the last token of the family name, then concatenate the first character of the given name with the last token of the family name.
   - If the last token of the family name is a single initial, then concatenate that character with the first token of the given name
4. Trim the cluster key to a maximum of 12 characters.

The following table provides some examples.

**Table 5-9 First and Last Name Cluster**

| dnGivenNames | dnFamilyName | dnClusterFirstLast |
|---|---|---|
| MARTIN | JONES | MJONES |
| MARTIN PETER | JONES | MJONES |
| MARTIN | MORGAN JONES | MJONES |
| JONES | M | MJONES |

## 5.2.6 Original Script Name (dnClusterOriginalScript)

The Original Script Name cluster provides a clustering method for matching names represented in non-Latin writing systems. The cluster builder generates a key for each token in the name.

> **Note**
>
> A single cluster value of "Myanmar" is generated for original script names written in the Burmese alphabet irrespective of the name. This is needed because token splitting is not possible for the Myanmar writing system as it does not use a space character between words. As a result, all original script names in Burmese script will be compared during matching. This should not cause performance issues during screening providing there are a low number of customer records using this writing system.

The default logic of the cluster builder is as follows:

1. Split the original script name into several name tokens, using a space character as the delimiter.
2. Trim each name token to a maximum of 5 characters.
3. Concatenate all of the trimmed token values with a pipe separator.
4. Deduplicate the list of keys.

The following table provides some examples.

**Table 5-10 Original Script Name Cluster**

| dnOriginalScriptName | dnClusterOriginalScript |
|---|---|
| Іван Антонавіч Шчурок | Іван \| Антон \| Шчуро |
| 林 紹 蕙 | 林\|紹\|蕙 |
| သင် သင် ဆွေ | Myanmar |
| محمد محمد منشوري | منشور\|محمد |

## 5.2.7 Entity Name Tokens (dnClusterNameTokens)

This cluster uses the standardized entity name to generate cluster keys.

The default logic is as follows:

1. Remove initials.
2. Remove common name tokens, such as Limited, or Corporation.
3. Normalize whitespace.
4. Convert space characters to pipe characters.

The following table provides some examples.

**Table 5-11 Entity Name Tokens Cluster**

| dnEntityName | Name with Initials and Common Name Tokens Stripped | dnClusterNameTokens |
|---|---|---|
| ANGLO CARIBBEAN CO LTD | ANGLO CARIBBEAN | ANGLO \| CARIBBEAN |
| GUAMATUR S A | GUAMATUR | GUAMATUR |

## 5.2.8 Entity Name Meta (dnClusterLongName)

This cluster uses the standardized entity name to generate cluster keys.

The default logic is as follows:

1. Remove initials.
2. Remove common name tokens, such as Limited, or Corporation.
3. Normalize whitespace.
4. Remove common business words, such as Company, or Association.
5. Transliterate any non-Latin characters into Latin.
6. Apply the Metaphone transformation (the standard double-Metaphone algorithm) outputting a key with a length of up to eight characters.

The following table provides some examples.

**Table 5-12 Name Metaphone Cluster**

| dnEntityName | Name with Initials and Common Name Tokens Stripped | dnClusterLongName |
|---|---|---|
| HAVANA INTERNATIONAL BANK LTD | HAVANA BANK | HFNPNK |
| CIMEX S A | CIMEX | SMKS |
| LA EMPRESA CUBANA DE FLETES | EMPRESA CUBANA FLETES | AMPRSKPN |

## 5.2.9 Entity Name Trim (dnClusterShortName)

This cluster uses the standardized entity name to generate cluster keys.

The default logic is as follows:

1. Remove all whitespace.
2. Left-trim the value to a maximum of 4 characters.

The following table provides some examples.

**Table 5-13 Name Trimmed Cluster**

| dnEntityName | dnClusterShortName |
|---|---|
| HAVANA INTERNATIONAL BANK LTD | HAVA |
| CIMEX S A | CIME |
| LA EMPRESA CUBANA DE FLETES | LAEM |

## 5.2.10 Entity Start End Name Tokens (dnClusterStartEndNameTokens)

This clustering method is designed as a looser version of the Entity Name Tokens cluster and allows for variation in entity names by creating clusters for the first five and last five characters of each name token.

The default logic is as follows:

1. Remove initials.
2. Remove common name tokens, such as Limited, or Corporation.
3. Normalize whitespace.
4. For each token that is longer than five characters, replace with two new tokens that are:
   - The first five characters of the token
   - The last five characters of the token

The following table provides some examples.

**Table 5-14    Start/End Name Tokens Cluster**

| dnEntityName | Name with Initials and Common Name Tokens Stripped | dnClusterStartEndNameTokens |
|---|---|---|
| HAVANA INTERNATIONAL BANK LTD | HAVANA INTERNATIONAL BANK | HAVAN\| AVANA\| INTER\| IONAL\| B ANK |
| CIMEX S A | CIMEX | CIMEX |
| LA EMPRESA CUBANA DE FLETES | LA EMPRESA CUBANA FLETES | LA\| EMPRE\| PRESA\| CUBAN\| UBA NA\| FLETE\| LETES |

## 5.3 Matching

Individual and entity matching is centered on individual and entity names respectively. Other items of data, such as associated countries and cities, are used to strengthen a possible match.

Match rule groups are places in the following order:

- Individual name match groups
- Aircraft name match groups
- Vessels name match groups
- Entity name match groups

The following general notes describe the approach to matching:

- Matches are ranked according to how well the name matches. An exact name match rates as a match at the highest level, with the lowest level being represented by two loosely possible name matches with a different name structure. Further ranking is imposed by how well additional information (such as city or country information, and date of birth information) matches between the records.
- TBAML allows for various levels of name match, including, but not limited to:
  - Name variation recognition. This is carried out by name standardization. For example, all variations of Mohammed (Muhamad, Mohammad, Mohamed and so on) are substituted with ‘Mohammed’ when matching. This is particularly used for given names, though also applied when matching whole names. For example, more than 20 variations of the name ‘Mohammed’ are recognized and considered to be the same name.
  - Allowances for name abbreviation and initials. For example, ‘Pete’ is a possible match to ‘Peter’, and ‘J’ is a possible match to ‘John’.
  - Allowances for typographical errors and transliteration differences. For example, ‘Abdool’ is a possible match to ‘Abdul’, even if the variants are not standardized.
  - Allowances for names being out of order or structured differently. For example, ‘Mohammed Abbas Al-Tikriti’ can be matched with ‘Mohammed Al-Tikriti Abbas’.
  - Allowance for additional names. For example, ‘Juan Carlos Ferreira’ can be matched with ‘Juan Ferreira’.
  - Allowance for names being split differently. For example, ‘Xiao Jian’ is a match to ‘Xiaojian’.

- TBAML attempts to prevent false positives by various means, including, but not limited to, the following methods:
  - Backing up typo tolerance with Metaphone matching. For example, ‘Mary’ and ‘Mark’ are not considered a match, although they are only one character different.
  - Backing up typo tolerance with consideration of the percentage of characters that are different. For example, the initials ‘A’ and ‘E’ are not considered a match, even though they are only one character different.
  - Considering the different significance and commonality of name tokens. For example, if name qualifiers such as ‘Al’ are shared between two Arabic names, this is not as significant as if an uncommon name such as ‘Abbas’ is shared.

> **Note**
>
> It may be advisable to tune the set of match rules that are activated. In particular, you may wish to activate or deactivate some of the lower match rules in the list, which lead to the weakest name matches. Factors affecting the usefulness of these rules include:
>
> - The policies of the organization.
> - The quality of the transaction data.
> - The provenance of the transaction data.

For example, Asian and Arabic names may be subject to more typographical and name ordering issues than other names. Where the data contains many of these names, the lower strength rules may identify more possible matches. The organization may want to review some or all of these as a matter of policy, or it may consider the matches too weak to review.

The required rules are easily activated or deactivated as needed in TBAML.

## 5.3.1 Match Rules

There are several different types of match rule involved in the name and address screening:

TBAML includes the following name and address screening match rules:

- The name matching rules: These are organized by the level of name match, with the strongest name matching rules placed at the top of the decision table.

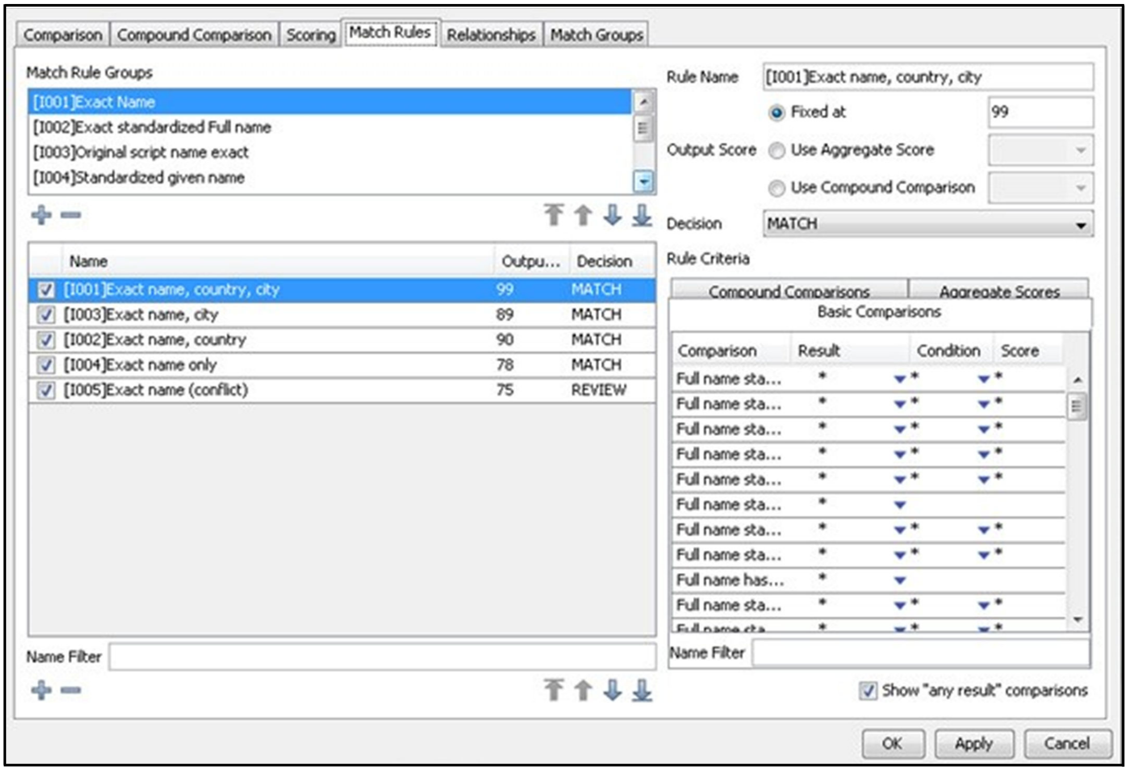
  > **Note**
  >
  > This means that the match rules are not ordered by strength across all identifiers. For example, a weaker name match that is strengthened by matches on city and country is likely to be a stronger overall match than a strong name with strongly contradictory data in the other fields.

  The last rule in each set is a 'conflict' rule, and in many cases will be disabled by default. These rules allow records which fulfill the specified level of name match but have conflicting supporting data fields indicating that a true match is unlikely.
- The loose name matching rules: These are also based around name matching, but identify looser matches and are not enabled by default. These rules are likely to result in a large

number of false positive matches and are most likely to be of use when screening against sanctions lists, where it is important that no true matches are missed.

For the sake of clarity, match rules are divided into groups. As each group is selected, the match rules it contains are displayed in the Match Rules tab:

**Figure 5-1 Match Rules Tab**



Match Rule Groups

- [I001]Exact Name
- [I002]Exact standardized Full name
- [I003]Original script name exact
- [I004]Standardized given name

|  | Name | Outpu... | Decision |
|---|---|---|---|
| ☑ | [I001]Exact name, country, city | 99 | MATCH |
| ☑ | [I003]Exact name, city | 89 | MATCH |
| ☑ | [I002]Exact name, country | 90 | MATCH |
| ☑ | [I004]Exact name only | 78 | MATCH |
| ☑ | [I005]Exact name (conflict) | 75 | REVIEW |

Name Filter

Rule Name: [I001]Exact name, country, city

Output Score: ● Fixed at 99
○ Use Aggregate Score
○ Use Compound Comparison

Decision: MATCH

Rule Criteria

| Comparison | Result | Condition | Score |
|---|---|---|---|
| Full name sta... | * | * | * |
| Full name sta... | * | * | * |
| Full name sta... | * | * | * |
| Full name sta... | * | * | * |
| Full name sta... | * | * | * |
| Full name sta... | * |  |  |
| Full name sta... | * | * | * |
| Full name sta... | * | * | * |
| Full name has... | * |  |  |
| Full name sta... | * | * | * |

Name Filter

☑ Show "any result" comparisons

The priority of the groups can be changed using the arrows below the Match Rules Group list. When a group is highlighted:

- Click ↑ to move it up one place in the list.
- Click ↓ to move it down one place in the list.
- Click ⤒to move it to the top of the list.
- Click ⤓to move it to the bottom of the list.

The remainder of this section describes the matching rules that are present in TBAML in greater detail.

## 5.3.2 Individual Name Matching Rules

The following are the individual name matching rules:

### Table 5-15 Individual Name Matching Rules

| Group Code | Matching Rule | Logic Summary | Example Matching Data |
|---|---|---|---|
| I001 | Exact name | Full name match after name standardization using full name map. | - |
| I002 | Exact standardized Full name | Given names and family name match exactly. | JOSEPH TSANGA<br>JOSEPH T'SANGA |
| I003 | Original script name exact | The original script Name fields match exactly. | АЛЕКСАНДР ОСОКИН<br>АЛЕКСАНДР ОСОКИН |
| I004 | Standardized given name | Given names match after name standardization using Given name map. Family name matches exactly. | BILL JONES WILLIAM JONES |
| I005 | Full name | The full name matches exactly, after standardization of all name tokens using the Given Name Map. | JOHN MIKE SMITH<br>JOHN MICHAEL SMITH |
| I006 | Full name without titles | The full name matches exactly, after standardization of all name tokens using the Given Name Map and removal of titles. | DR DOUGLAS BAKER<br>DOUGLAS BAKER |
| I007 | Abbreviated standardized given name | Given names match using a Starts With comparison, after name standardization using the Given Name Map. Family name matches exactly. | JOSEPH ABANDA TSANGA<br>JOSEPH T'SANGA |
| I008 | Given name similar and sounds like | Given name matches with an Edit Distance of 1 or 2 after name standardization. At least one of the given names, excluding initials, must match by a 4-character Metaphone key. Family name matches exactly | JOSEPH ABANDA<br>JOESPH ABANDA |
| I009 | First name similar and sounds like | The first given name matches with an Edit Distance of 1 or 2 and with a Character Match Percentage of 66% or more, after given name standardization. At least one of the given names, excluding initials, must match by a 4-character Metaphone key. Family name matches exactly. | AMER MOHAMMAD RASHEED<br>AL UBAIDI<br>AMIR RASHID MOHAMMED<br>AL UBAIDI |

**Table 5-15 (Cont.) Individual Name Matching Rules**

| Group Code | Matching Rule | Logic Summary | Example Matching Data |
|---|---|---|---|
| I010 | Additional given names | All name tokens from the given names field with fewest tokens must be present in the other given names field. Family name matches exactly | MOHAMMED<br>HANIF<br>DIN MOHAMED<br>HANIF |
| I011 | Additional names | All name tokens from the full name with fewest tokens must be present in the other full name. At least 2 name tokens must match with the same matching logic; that is, if a name only has one token it is not considered a match. At least 2 name tokens must exist in the Full Name.<br>**Note:** Word Match Count may return >1 if a single name matches twice in a longer name string. For example, 'ABDUL' matches 'ABDUL ABDUL' with a Word Match Count of 2. Matching is order sensitive. | LOTFI RIHANI<br>LOTFI BEN ABDUL HAMID BEN ALI RIHANI |
| I012 | Original script name in any order | All names in the original script name fields match, regardless of order. | Καρλος<br>Μολινα<br>Μολινα<br>Καρλος |
| I013 | Original script name with typos | Original script name fields match with an 80% + Character Match Percentage score. | Καρλος<br>Μολινα<br>Καρλος<br>Μολιννα |
| I014 | All names in any order | All names in the full name match (using a Word Edit Distance of 0) after name token standardization, in any order. A single typo (1 character edit) is allowed in each name token. | ABDUL JABBER OMARI<br>OMARI ABDUL JABBER |
| I015 | Abbreviated given name | Given names match using a Starts With comparison. Family name is a close metaphone match. | CHRIS<br>HUNT<br>CHRISTOPHER<br>HUNTER |

**Table 5-15 (Cont.) Individual Name Matching Rules**

| Group Code | Matching Rule | Logic Summary | Example Matching Data |
|---|---|---|---|
| I016 | Abbreviated given name and family name typos | Given names match using a Starts With comparison, after name standardization using Given Name Map. Family name matches with an edit difference of 1-2. At least one of the family name tokens, excluding initials must match by a 4-character Metaphone key. | IBRAHIM ABDUL SALAM MOHAMED BOYASSEER<br>IBRAHIM<br>BOYASEER |
| I017 | Abbreviated given name without titles and family name with typos | The first given name matches with a Starts With match, after name token standardization and stripping titles. Family name matches with an edit difference of 1-2. At least one of the family name tokens, excluding initials, must match by a 4-character Metaphone key. | SAHIR BARHAN<br>DR SAHIR MUSA<br>BERHIN |
| I018 | Original script name in any order with typos | All names in the original script name fields match, regardless of order, with each name requiring an 80%+ Character Match Percentage score. | Хасан Ченгић<br>ЧенгићХассан |
| I019 | First name and full name similar and sounds like | The full name matches with a Character Match Percentage of 80% or above, after name token standardization. At least one of the family name tokens, excluding initials, must match by a 4-character Metaphone key. | MOHAMMAD HUSAYN MASTASAEED<br>MOHAMMAD HASSAN<br>MASTASAEED |
| I020 | Given name similar and family names and sounds like | The given name matches with an Edit Distance of 1 or 2, after name standardization. The given name matches by 4-character Metaphone key, after name standardization. The family name matches with an Edit Distance of 1-2. The family name matches by 4-character Metaphone key. | AMER MOHAMMAD RASHEED<br>AL UBAIDI<br>AMIR RASHID MOHAMMED<br>AL UBEIDI |

**Table 5-15 (Cont.) Individual Name Matching Rules**

| Group Code | Matching Rule | Logic Summary | Example Matching Data |
|---|---|---|---|
| I021 | Abbreviated given name and family name similar | The first given name matches with a Starts With match, after name token standardization. The family name matches with an Edit Distance of 1 or 2. The family name matches by 4-character Metaphone key. | VIKTOR ANATOLYEVICH BOUT<br>VICTOR<br>BOOT |
| I022 | Full Name no whitespace | Combination of Given name an Family name without spaces | CHRIS HUNT<br>CHRISTOPHER<br>HUNTER |
| I023 | Original script name additional names | All names in one original script name field must be fully contained within the other field, provided there are at least two names in each field. | Миленко Врачар<br>Миленко Иванович Врачар |
| I024 | Additional names typo tolerant | All name tokens from the full name with fewest tokens must be present in the other full name. A character error tolerance of 20% is allowed (that is, one character edit every 5 characters). At least 2 name tokens must match with the same matching logic. If a name contains only one token it is not considered a match according to this rule.<br>**Note:** Word Match Count may return >1 if a single name matches twice in a longer name string. For example, ‘ABDUL’ matches ‘ABDUL ABDUL’ with a Word Match Count of 2. Matching is order sensitive. | ABDUL WAHED SHAFIQ<br>ABDUL WAHAD |
| I025 | Full name contained and multiple names in common | The full name matches with a Contains match, after standardization of all name tokens using the Given Name Map. At least 2 name tokens must match in the full name. | ABU BAKAR<br>ABU BAKAR BA’ASYI |

**Table 5-15 (Cont.) Individual Name Matching Rules**

| Group Code | Matching Rule | Logic Summary | Example Matching Data |
|---|---|---|---|
| I026 | Full name characters longer | The full name matches with a Longest Common Substring Sum Percentage of 90%+, relating to the longer string, and considering substrings of 5 characters or more in length, after name standardization. | MOHAMMED AL GHABRA ALGHABRA MUHAMAD<br>RAMATULLAH WAHIDYAR FAQIR MOHAMMAD<br>WAHIDYAR RAMA TULLAH |
| I027 | Original script name additional names with typos | All names in one original script name field must be fully contained within the other field, provided there are at least two names (all of which have an 80%+ Character Match Percentage) in each field. | ЮРИ НЕЁЛОВ<br>Юрий Васильевич Неёлов |
| I028 | Abbreviated first name | The first given name matches with a Starts With match, after name token standardization. Family name matches exactly. | KHADAF ABUBAKAR JANJALANI<br>KHADAFFI<br>JANJALANI |
| I029 | Additional names in any order | All name tokens from the full name with fewest tokens must be present in the other full name. At least 2 name tokens must match with the same matching logic. If a name contains only one token it is not considered a match according to this rule.<br>**Note:** Word Match Count may return >1 if a single name matches twice in a longer name string. For example, ‘ABDUL’ matches ‘ABDUL ABDUL’ with a Word Match Count of 2. Matching is order insensitive. | HA THI NGUYEN<br>THI HA |

**Table 5-15 (Cont.) Individual Name Matching Rules**

| Group Code | Matching Rule | Logic Summary | Example Matching Data |
|---|---|---|---|
| I030 | Additional names in any order typo tolerant | All name tokens from the full name with fewest tokens must be present in the other full name. A character error tolerance of 20% is allowed (that is, one character edit every 5 characters). At least 2 name tokens must match with the same matching logic. If a name contains only one token it is not considered a match according to this rule.<br>**Note:** Word Match Count may return >1 if a single name matches twice in a longer name string. For example, 'ABDUL' matches 'ABDUL ABDUL' with a Word Match Count of 2. Matching is order insensitive. | STEPHENS MARTIN<br>MARRTIN JOHN STEPHENS |

## 5.3.3 Loose Individual Name Matching Rules

The following table lists the loose individual name matching rules:

**Table 5-16 Loose Individual Name Matching Rules**

| Group Code | Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| I031 | Full name characters shorter only | The full name matches with a Longest Common Substring Sum Percentage of 90%, relating to the shorter string, and considering substrings of 5 characters or more in length, after name standardization. At least 2 name tokens must exist in the full name. | ABU BAKAR<br>ABU BAKAR BA'ASYI |

**Table 5-16 (Cont.) Loose Individual Name Matching Rules**

| Group Code | Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| I032 | Full name no initials match with initials in any order relating to shorter. | All initials in one Full Name field must be fully contained within the initials of the other Full Name field;<br>AND the standardized Full Name field without initial must be fully contained within the other standardized Full Name field without initials;<br>AND both fields must contain at least two names. | CARL J FISHER<br>J C FISHER |
| I033 | Full name contained, last initial same, primary list is single token. | The Full Name field from the watch list record contains only one name, which is fully contained within the record being screened;<br>AND the initial of the last name in the record being screened must match the initial of the name in the watch list record. | JANINE CHERRY<br>CHERRY |

## 5.3.4 Aircraft Matching Rules

The following table details the Aircraft matching rules:

**Table 5-17 Aircraft Matching Rules**

| Group Code | Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| A001 | Aircraft part-standardized name exact | The part-standardized entity name matches the name of a listed aircraft exactly | |
| A002 | Aircraft name exact | The entity name matches the name of a listed aircraft after number cardinal and ordinal standardization | 4TH YK-AYF<br>FOURTH YK-AYF |
| A003 | Aircraft part-standardized name with typos | The part-standardized entity name matches the name of a listed aircraft with a Character Match Percentage of 80-99%. | N840PN 1 N840PN |

**Table 5-17 (Cont.) Aircraft Matching Rules**

| Group Code | Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| A004 | Aircraft name with typos | The entity names match with a Character Match Percentage of 80-99% after number cardinal and Ordinal standardization | |

## 5.3.5 Vessel Matching Rules

The following table details the Vessel matching rules:

**Table 5-18 Vessel Matching Rules**

| Group Code | Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| V002 | Vessel name exact | The entity name matches the name of a listed vessel after number cardinal and ordinal standardization | 4th OCEAN<br>FOURTH OCEAN |
| V003 | Vessel part-standardized name with typos | Thepart-standardized entity name matches the name of a listed vessel with a Character Match Percentage of 80-99% | RAHIM RAHIM 3 |
| V004 | Vessel name with typos | The entity name matches the name of a listed vessel after number cardinal and ordinal standardization | RAHUM 3<br>TRAHIM THREE |
| V001 | Vessel part-standardized name exact | The part-standardized entity name matches the name of a listed vessel exactly. | DYNASTY DYNASTY |

## 5.3.6 Entity Matching Rules

The match rules in Oracle Financial Services Trade-Based Anti Money Laundering are organized by the level of entity name match, with the strongest name matching rules at the top of the decision table.

There are two types of matching rules involved in entity screening:

- Entity name matching rules: Entity name matching rules are organized by the level of entity name match, with the strongest matching rules placed at the top of the decision table. This means that the match rules are not ordered by strength across all identifiers. For example, a weakermatch rule that is strengthened by matches on City and Country is likely to be a stronger overall match than a strong match rule with strongly contradictory data in the other fields.

- Loose entity matching rules: These are also based around entity name matching, but identify looser matches and are not enabled by default. These rules are likely to result in a large number offalse positive matches and are most likely to be of use when screening against sanctioned lists where it is important that no true matches are missed.

For the sake of clarity, match rules are divided into groups, as shown in the following tables.

**Note**

All entity matching rules use a standardized form of the entity name. The strongest rules use the 'part-standardized name', meaning the entity names match after only simple global standardizations (such as considering AND and & as the same) are applied. Other rules apply additional rules for standardization as noted in the following table. Usage of 'word' in the table implies a space-delimited token in the prepared names.

The following table shows the Entity Name Matching Rules:

**Table 5-19 Entity Name Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E001 | Part- standardized name exact | The part-standardized entity name matches a listed entity name exactly. | HUMAN APPEAL INTERNATIONAL<br>HUMAN APPEAL INTERNATIONAL |
| E002 | Name exact | The entity names match exactly after number cardinal and ordinal standardization. | ABN Bank ABN |
| E003 | Original script name exact | The original script names match exactly. | НИАЭП ОАО<br>НИАЭП ОАО |
| E004 | Name without suffixes exact | The entity names match exactly after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. | CAPITAL DIRECT LTD<br>CAPITAL |
| E005 | Name without business words similar and sounds like | The entity names match with a Word Match Percentage of 80% after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The first word of each name has the same 4-character Metaphone key. | PARAGON INVESTMENT CORPORATION<br>PIC |

**Table 5-19 (Cont.) Entity Name Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E006 | Name without business words exact | The entity names match exactly after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. | LIFE HEALTHCARE GROUP HOLDINGS LTD<br>LHCG |
| E001 | Part- standardized name exact | The part-standardized entity name matches a listed entity name exactly. | HUMAN APPEAL INTERNATIONAL HUMAN APPEAL INTERNATIONAL |
| E007 | Name without business words has all words out-of-order | All remaining words in each entity name match exactly, but in any order, after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. | HEALTH EDUCATION SERVICES HEALTH SERVICES |
| E008 | Name without suffixes starts with and multiple names in common | The entity names are a Starts With match after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. There are at least two significant words (not common business words) in common between the two names. The listed name is not an acronym alias of a longer primary entity name. | BAE SYSTEMS (LANCASTER HOUSE) LIMITED<br>BAE SYSTEMS LIMITED |
| E009 | Name without business words has all words with typos | All remaining words in each entity name match with a Character Match Percentage of 80 or more, after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed | GERBERA ASSOCIATES LTD GERBARA ASSOCIATES LTD |
| E010 | Original script name in any order | All words in the Original Script Names match exactly, in any order | НИАЭП ОАО<br>НИАЭП ОАО |

**Table 5-19 (Cont.) Entity Name Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E011 | Original script name with typos | The Original Script Names match with a Character Match Percentage of 80% or more. | НИАЭП ОАО<br>НИАЭП ОАО |
| E012 | Name without business words with typos, and sounds like | The entity names match with a Character Match Percentage of 80 ore more after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The first word of each name has the same 4-character Metaphone key and the first three letters of each name are the same. | GOLDSTREAMPROPERTIES LTD<br>GOLDSTREAMPROPERTIES LTD |
| E013 | Name without suffixes contains, similar and multiple names in common | The entity names are a Contains match and the Word Edit Distance is no more than one between the names (where each word matches with a Character Match Percentage of 80 or more), after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. There are at least two significant words (not common business words) in common between the two names. | HAMPSHIRE HERITAGE DEVELOPMENTS LTD<br>HAMPSHIRE HERITAGE DEVELOPMENTS LTD |

**Table 5-19 (Cont.) Entity Name Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E014 | Name has additional words, sounds like and multiple names in common | All words in the shorter entity name exist in the longer entity name (in order) after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. There are at least two significant words (not common business words) in common between the two names. The list name is not an acronym alias of a longer primary entity name. | MOSCOW CITY CENTER PLC MOSCOW CITY CENTER PLC |
| E015 | Name without business words contains, sounds like and multiple names in common | The entity name is a Contains match with a listed entity name, after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. There are at least two significant words (not common business words) in common between the twonames. The first word of each name has the same 4-character Metaphone key. | HI-TECH RECRUITMENT LTD HI-TEC RECRUITMENT LTD |
| E016 | Original script name in any order with typos | All words in the original script name match with a Character Match Percentage of 80or more, in any order. | НИАЭП ОАО<br>НИАЭП ОАО |
| E017 | Name without business words has most words out-of-order | The entity names match (in any order) with a Word Match Percentage of between 75 and 99, after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The list name is not an acronym alias of a longer primary entity name. | BACK TO HEALTH CLINICS LIMITED<br>BACK TO HEALTH CLINICS LIMITED |

Table 5-19 (Cont.) Entity Name Matching Rules

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E018 | Name without business words, similar, sounds like, with multiple names and a residual token in common.<br>**Note:** The group name differs from the rule name. | All words in the shorter entity name exist in the longer entity name (in order) after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. There are at least two significant words (not common business words) in common between the two names, andat least one of these is not a word in the English dictionary or a very common word in Watchlist name data. The list name is not an acronym alias of a longer primary entity name. | CHARLES F ASH CONSTRUCTION CO INC CHARLES F ASH CONSTRUCTION CO INC |
| E019 | Name without business words, similar with typos, sounds like, with multiple names and residual token in common.<br>**Note:** The group name differs from the rule name. See the Match dialog for details. | All words in the shorter entity name match with a Character Match Percentage of 80 or more in the longer entity name (in order) after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. There are at least two significant words (not common business words) that match with a Character Match Percentage of 80 or more, and at least one of these is not a word in the English dictionary or a very common word in Watchlist name data. The list name is not an acronym alias of a longer primary entity name. | CLARKS HOME BAKERY LTD CLARKS HOME BAKERY LTD |

**Table 5-19 (Cont.) Entity Name Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E020 | Name has additional words tolerant, sounds like and multiple names in common | All words in the shorter entity name match in the longer entity name (in order) with a Character Match Percentage of 80 or more after number cardinal and ordinal standardization. There are at least two significant words (not common business words) in common between the two names. The list name is not an acronym alias of a longer primary entity name. | ABU AL FULUS ABU AL FULUS |
| E021 | Name without suffixes contains, similar and residual token in common | The entity names are a Contains match and the Word Edit Distance is no more than one between the names (where each word matches with a Character Match Percentage of 80 or more), after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. There is at least one significant word in common (not a common business word, a word in the English dictionary or a very common word in Watchlist name data). | ACCLAIM ACM LTD ACCLAIM ACM LTD |

**Table 5-19 (Cont.) Entity Name Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E022 | Name without suffixes starts with and residual token in common | The entity names are a Starts With match after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. There is at least one significant word in common (not a common business word, a word in the English dictionary or a very common word in Watchlist name data). The listed name is not an acronym alias of a longer primary entity name. | ENRON METALS BROKERS LTD ENRON METALS BROKERS LTD |
| E023 | Name without suffixes starts with and substring in common | The entity names are a Starts With match, and there is a common substring at least 8 characters in length, after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. The listed name is not an acronym alias of a longer primary entity name. | ACCURATE SECTION BENDERS LTD<br>ACCURATE SECTION BENDERS LTD |
| E024 | Name without suffixes contains, residual token in common and significant overlap | The entity names are a Contains match and the Word Match Percentage is 50 or more, after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. There is at least one significant word in common (not a common business word, a word in the English dictionary or a very common word in Watchlist name data). | NON EMERGENCYTRANSPORT INC<br>NON EMERGENCYTRANSPORT INC |

**Table 5-19 (Cont.) Entity Name Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E025 | Name without common tokens exact, and multiple residual tokens in common | The entity names match exactly, with at least two words matching, after number cardinal and ordinal standardization, and after common company prefixes, suffixes, and other words, and all English dictionary and common Watchlist name words are removed. | LIFE CARE CENTER PUNTA GORDA<br>LIFE CARE CENTER PUNTA GORDA |
| E026 | Original script name has additional names | All words in the shorter original script name match in the longer original script name (in order), and there are at least two matching words. | НИАЭП ОАО<br>НИАЭП ОАО |
| E027 | Name without suffixes contains, multiplenames in common and significant overlap | The entity names are a Contains match and the Word Match Percentage is 50 or more, after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. There is at least two significant words (not common business words) that match with a Character Match Percentage of 80 or more. | CAPITAL CITY TRANS SERV INC CAPITAL CITY TRANS SERV INC |

**Table 5-19 (Cont.) Entity Name Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E029 | Name without business words similar with typos,sounds like and significant overlap | All words in the shorter entity name match with a Character Match Percentage of 80 or more in the longer entity name (in order) after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The names match with a Word Match Percentage of 50 or more when common business words are not stripped. There are at least two significant words (not common business words) that match with a Character Match Percentage of 80 or more.<br>The first word of each name has the same<br>4-character Metaphone key. The list name is not an acronym alias of a longer primary entity name. | MED AMERICA CLINICS INC MED AMERICA CLINICS INC |
| E030 | Name has additional words, sounds like and residual token in common | All words in the shorter entity name exist in the longer entity name (in order) after number cardinal and ordinal standardization. There is at least one significant word (not a common business word, an English dictionary word or a word or a common Watchlist name word) in common between the two names. The list name is not an acronym alias of a longer primary entity name. | DJ CASE AND ASSOCIATES INC DJ CASE AND ASSOCIATES INC |

Table 5-19 (Cont.) Entity Name Matching Rules

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E031 | Name has additional words with typos, sounds like and residual token in common | All words in the shorter entity name match with a Character Match Percentage of 80 or more in the longer entity name (in order) after number cardinal and ordinal standardization. There is at least one significant word (not a common business word, an English dictionary word or a word or a common Watchlist name word) that matches with a Character Match Percentage of 80 or more. The list name is not an acronym alias of a longer primary entity name. | GARLICK HELICOPTERS INC<br>GARLICK HELICOPTERS INC |
| E028 | Name without business words similar and full name sounds like | The entity names match with a Character Match Percentage of between 80 and 99 after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The names share the same metaphone key after number cardinal and ordinal standardization. | IBERIA AIRLINES<br>IBERIAN AIRLINES |
| E032 | Name has additional words, sounds like and substring in common | All words in the shorter entity name exist in the longer entity name (in order) after number cardinal and ordinal standardization. There is a common substring of at least 8 characters in length between the two names after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The list name is not an acronym alias of a longer primary entity name. | NATIONWIDE SECRETARIAL SERVICES LTD<br>NATIONWIDE SECRETARIAL SERVICES LTD |

**Table 5-19 (Cont.) Entity Name Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E033 | Name without business words, similar, sounds like and multiple names in common | All words in the shorter entity name match in the longer entity name (in order) after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. There are at least two significant words (not common business words) that match. The first word of each name has the same 4-character Metaphone key. The list name is not an acronym alias of a longer primary entity name. | CENTRAL OKLAHOMA FAMILY MEDICAL CENTER<br>CENTRAL OKLAHOMA FAMILY MEDICAL CENTER |
| E034 | Name without business words, similar with typos,sounds like and multiple names in common | All words in the shorter entity name match with a Character Match Percentage of 80 or more in the longer entity name (in order) after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. There are at least two significant words (not common business words) that match with a Character Match Percentage of 80 or more. The first word of each name has the same 4-character Metaphone key. The list name is not an acronym alias of a longer primary entity name. | BLACK WORLD COLLEGEOF HAIR DESIGN<br>BLACK WORLD COLLEGEOF HAIR DESIGN |

**Table 5-19 (Cont.) Entity Name Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E035 | Name without business words has typos and sounds like | The entity names match with a Character Match Percentage of between 80 and 99 after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The first word of each name has the same 4-character Metaphone key. | BOURNE CHIROPRACTIC LTD<br>BOURNE CHIROPRACTIC LTD |
| E036 | Name without suffixes contains with typos and multiple names in common | The entity names are a 'Contains' match where each word matches with a Character Match Percentage of 80 or more after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. There are at least two significant words (not common business words) that match. | MEDICAB OF METRO NEW ORLEANS<br>MEDICAB OF METRO NEW ORLEANS |
| E037 | Name without suffixes contains, similar, and multiple words in common | The entity names are a Contains match and the Word Edit Distance is no more than one between the names (where each word matches with a Character Match Percentage of 80 or more), after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. There are at least two significant words (not common business words) that match with a Character Match Percentage of 80 or more. | GROSVENORNURSING SERVICES<br>GROSVENOR NURSING SERVICES |

Table 5-19 (Cont.) Entity Name Matching Rules

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E038 | Original script name has additional names with typos | All names in one original script name field must be fully contained within the other field, provided there are at least two names (all of which have an 80%+ Character Match Percentage) in each field. | НИАЭП ОАОНИАЭП ОАО |
| E039 | Name has additional words and sounds like | All words in the shorter entity name exist in the longer entity name (in order) after number cardinal and ordinal standardization. | ATRIUM INCORPORATORS WORLDWIDE LTD<br>ATRIUM INCORPORATORS<br>WORLDWIDE LTD |
| E040 | Name has additional words with typos and sounds like | All words in the shorter entity name match in the longer entity name (in order) with a Character Match Percentage of 80 or more<br>after number cardinal and ordinal<br>standardization. The first word of each name has the same 4-character Metaphone key. | BRILLIANT GENERALBUILDING CONTRACTOR LTD<br>BRILLIANT GENERALBUILDING CONTRACTOR LTD |
| E041 | Name without business words loose match and full name sounds like | The entity names match with a Character Match Percentage of between 60 and 79 after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The names have the same Metaphone key. | BRC BRC |

**Table 5-19 (Cont.) Entity Name Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E042 | Name without business words contains, sounds like, and residual token in common | Theentity names are a Contains match after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. There is at least one significant word (not a common business word, and English dictionary word or a very common word in Watchlist name data) in common between the two names. The first word of each name has the same 4- character Metaphone key. The list name is not an acronym alias of a longer primary entity name. | GARLICK HELICOPTERS INC GARLICK HELICOPTERS INC |
| E043 | Name without business words contains, sounds like, and substring in common | The entity names match with a Character Match Percentage of between 60 and 79 after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The names have the same Metaphone key. | NATIONWIDE SECRETARIAL SERVICES LTD NATIONWIDE SECRETARIAL SERVICES LTD |
| E044 | Name without suffixes starts with | The entity names are a Starts With match after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. The list name is not an acronym alias of a longer primary entity name. | MARK MARK |

**Table 5-19 (Cont.) Entity Name Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E045 | Name without business words has additional words and sounds like | All words in the shorter entity name exist in the longer entity name (in order) after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The first word of each name has the same 4- character Metaphone key. The list name is not an acronym alias of a longer primary entity name. | CROSS CROSS |
| E046 | Name without business words has additional words with typos and sounds like | All words in the shorter entity name match with a Character Match Percentage of 80 or more in the longer entity name (in order) after number cardinal and ordinal standardization and after common company prefixes, suffixes and other words are removed.The first word of each name has the same 4-character Metaphone key. The list name is not an acronym alias of a longer primary entity name. | GROSVENORNURSING SERVICES GROSVENOR NURSING SERVICES |
| E047 | Name without business words contains and sounds like | The entity names are a Contains match after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The first word of each name has the same 4-character Metaphone key. The list name is not an acronym alias of a longer primary entity name. | ENRON METALS BROKERS LTD ENRON METALS BROKERS LTD |
| E048 | Name without suffixes starts with and allows acronyms | The entity names are a Starts With match after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. | ANR ANR |

**Table 5-19 (Cont.) Entity Name Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E049 | Name without suffixes contains, significant overlap and multiple words in common | The entity names are a Contains match, there are at least two words that match with a Character Match Percentage of 80 or more, and the two entity names match with a Word Match Percentage of 50 or more after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. | ALTAR ALTAR |
| E050 | Name contains with typos and multiple words in common | The entity names are a 'Contains' match where each word matches with a Character Match Percentage of 80 or more after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. There are at least two words (not prefixes or suffixes) that match. | CHERRY CHERRY |

The following table shows the Loose Entity Matching Rules:

**Table 5-20 Loose Entity Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E0001 | Name exact | The entity names match exactly after number cardinal and ordinal standardization | NOVEMBER SEVENTEEN<br>NOVEMBER 17 |
| E0002 | Original script name in any order | All words in the Original Script Names match exactly, in any order. | НИАЗП ОАОНИАЗП ОАО |
| E0003 | Original script name with typos | The Original Script Names match with a Character Match Percentage of 80% or more. | ΕΠΑΝΑΣΤΑΤΙΚΉ ΑΡΙΣΤΕΡΆ<br>ΕΠΑΝΑΣΤΑΤΙΚΉ ΑΡΙΣΤΕΡΆ |

**Table 5-20 (Cont.) Loose Entity Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E0004 | Original script name in any order with typos | All words in the original script name match with a Character Match Percentage of 80or more, in any order. | ΕΠΑΝΑΣΤΑΤΙΚΉ ΑΡΙΣΤΕΡΆ<br>ΕΠΑΝΑΣΤΑΤΙΚΉ ΑΡΙΣΤΕΡΆ |
| E0005 | Original script name has additional names | All words in the shorter original script name match in the longer original script name (in order), and there are at least two matching words. | ВЪОРЪЖЕНА ИСЛЯМСКА ГРУПА<br>ВЪОРЪЖЕНА ИСЛЯМСКА ГРУПА |
| E0006 | Original script name has additional names with typos | All words in the shorter original script name match in the longer original script name (in order) with a Character Match Percentage of 80 or more, and there are at least two matching words. | ВЪОРЪЖЕНА ИСЛЯМСКА ГРУПА<br>ВЪОРЪЖЕНА ИСЛЯМСКА ГРУПА |
| E0007 | Name without business words contains, sounds like, and residual token in common | The entity names are a Contains match after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. There is at least one significant word (not a common business word, and English dictionary word or a very common word in Watchlist name data) in common between the two names. The first word of each name has the same 4-character Metaphone key.<br>The list name is not an acronym alias of a longer primary entity name. | HENDERSON EQUITY PARTNERS GP LTD<br>HENDERSON EQUITY PARTNERS GP LTD |

**Table 5-20 (Cont.) Loose Entity Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E0008 | Name without business words contains, sounds like, and substring in common | The entity names are a Contains match and there is a common substring at least 8 characters in length after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The first word of each name has the same 4-character Metaphone key. The list name is not an acronym alias of a longer primary entity name. | HAMILTON NEWS<br>HAMILTON NEWS |
| E0009 | Name without suffixes starts with | The entity names are a Starts With match after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. The list name is not an acronym alias of a longer primary entity name. | JACOB JACOB |
| E0010 | Name without business words has additional words and sounds like | All words in the shorter entity name exist in the longer entity name (in order) after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The first word of each name has the same 4- character Metaphone key. The list name is not an acronym alias of a longer primary entity name. | IDEAL SOLUTION ESTATES MANAGEMENT LTD<br>IDEAL SOLUTION ESTATES MANAGEMENT LTD |

Table 5-20 (Cont.) Loose Entity Matching Rules

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E0011 | Name without business words has additional words with typos and sounds like | All words in the shorter entity name match with a Character Match Percentage of 80 or more in the longer entity name (in order) after number cardinal and ordinal standardization, and after commoncompany prefixes, suffixes and other words are removed. The first word of each name has the same 4-character Metaphone key. The list name is not an acronym alias of a longer primary entity name. | AVANT GARD LTD AVANT GARD LTD |
| E0012 | Name without business words contains and sounds like | The entity names are a Contains match after number cardinal and ordinal standardization, and after common company prefixes, suffixes and other words are removed. The first word of each name has the same 4- character Metaphone key. The list name is not an acronym alias of a longer primary entity name. | MOREX TRADING LTD MOREX TRADING LTD |
| E0013 | Name without suffixes starts with and allows acronyms | The entity names are a Starts With match after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. | INTERTRADE CLASSIC LTD INTERTRADE CLASSIC LTD |

**Table 5-20 (Cont.) Loose Entity Matching Rules**

| Group Code | Name Matching Rule | Summary of Rule Logic | Example Matching Data |
|---|---|---|---|
| E0014 | Name without suffixes contains, significant overlap and multiple words in common | The entity names are a Contains match, there are at least two words that match<br>with a Character Match Percentage of 80 or more, and the two entity names<br>match with a Word Match Percentage of 50 or more after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. | INTERTRADE CLASSIC LTD INTERTRADE CLASSIC LTD |
| E0015 | Name contains with typos and multiple words in common | The entity names are a 'Contains' match where each word matches with a Character Match Percentage of 80 or more after number cardinal and ordinal standardization, and after common company prefixes and suffixes are removed. There are at least two words (not prefixes or suffixes) that match. | MOREX TRADING LTD MOREX TRADING LTD |

# 6
# Trade Goods Matching

This chapter discusses the matching webservice used to extract the trade goods name and match the name against the prohibited goods list. This list provides country-wise data.

### Identifiers

The following identifiers are used for Trade Goods matching:

**Table 6-1 Trade Goods Identifiers**

| Identifier Description | Standard Prepared Attribute Name |
|---|---|
| Goods Name | Goods Name |
| Country From | Country From |
| Country To | Country To |
| Goods Synonyms | Goods Synonyms |

### Clusters

The following clusters are used for Trade Goods matching:

- GoodsName: This cluster uses the Goods Name to generate cluster keys. Goods Name Cluster allows new records to be matched against existing records in a system.
- GoodsSynonyms: This cluster uses Goods Synonyms to generate cluster keys. Goods Synonyms Cluster allows new records to be matched against existing records in a system.

## 6.1 Trade Goods Match Rules

The following match rules are used for Trade Goods match processing:

**Table 6-2 Trade Goods Match Rules**

| Rule Code | Matching Rule | Summary of Rule Logic | Example Matching Data Goods Name | Example Matching Data Import Country ISO from | Example Matching Data Export Country ISO to |
|---|---|---|---|---|---|
| G001 | Exact goods name, Import Country ISO from, Export Country ISO to | Exact Match against goods name, Country ISO (Import Prohibition from) AND Country ISO (Export Prohibition to) columns of prohibited goods reference data | Crude Oil | CH | DN |
| G002 | Exact goods name, Import Country ISO from | Exact Match against goods name, Country ISO (Import Prohibition from) columns of prohibited goods reference data | Crude Oil | CH | CR |
| G003 | Exact goods name, Export Country ISO to | Exact Match against goods name, Country ISO (Export Prohibition to) columns of prohibited goods reference data | Crude Oil | VE | UG |
| G004 | Exact goods name | Exact Match against goods name column ofprohibited goods reference data | Crude Oil | VE | UG |
| G005 | Exact goods synonym name, Import Country ISO from, Export Country ISO to | Exact Match against goods synonym name, Country ISO (Import Prohibition from) AND Country ISO (Export Prohibition to) columns of prohibited goods reference data | Oil | TW | GB |

**Table 6-2 (Cont.) Trade Goods Match Rules**

| Rule Code | Matching Rule | Summary of Rule Logic | Example Matching Data Goods Name | Example Matching Data Import Country ISO from | Example Matching Data Export Country ISO to |
|---|---|---|---|---|---|
| G006 | Exact goods synonym name, Import Country ISO from | Exact Match against goods synonym name, Country ISO (Import Prohibition from) columns of prohibited goods reference data | Oil | TW | US |
| G007 | Exact goods synonym name, Export Country ISO to | Exact Match against goods synonym name, Country ISO (Export Prohibition to) columns of prohibited goods reference data | Oil | UK | UM |
| G008 | Exact goods synonym name | Exact Match against goods synonym name column of prohibited goods reference data | Oil | UK | UM |
| G009 | Goods name in any order, Import Country ISO from, Export Country ISO to | Matches city ISO code exactly | Crude Oil Oil Crude | UR | SS |
| G0010 | Goods name in any order, Import Country ISO from | Match against goods name in any order exact Country ISO (Import Prohibition from) columns of prohibited goods reference data | Crude Oil Oil Crude | UR | SS |

**Table 6-2 (Cont.) Trade Goods Match Rules**

| Rule Code | Matching Rule | Summary of Rule Logic | Example Matching Data Goods Name | Example Matching Data Import Country ISO from | Example Matching Data Export Country ISO to |
|---|---|---|---|---|---|
| G011 | Goods name in any order, Export Country ISO to | Match against goods name in any order, exact Country ISO (Export Prohibition to) columns of prohibited goods reference data | Crude Oil Oil Crude | VN | CH |
| G012 | Goods name in any order | Match against goods name column in any order of prohibited goods reference data | Crude Oil Oil Crude | VN | CH |
| G013 | Goods name at least one word matching, Import Country ISO from, Export Country ISO to | Word match count is >0 against goods name in reference data, exact Country ISO (Import Prohibition from) AND exact Country ISO (Export Prohibition to) | Crude Oil | ZA | SW |
| G014 | Goods name atleast one word matching, Import Country ISO from | Word match count is >0 against goods name in reference data, exact Country ISO (Import Prohibition from) | Crude Oil | ZA | SW |
| G015 | Goods name at least one word matching, Export Country ISO to | Word match count is >0 against goods name in reference data, exact Country ISO (Export Prohibition to) | Crude Oil | TW | GB |
| G016 | Goods name at least one word matching | Word match count is >0 against goods name in reference data | Crude<br>Oil | TH | TJ |

# 7

# Trade Port Matching

This chapter discusses the matching webservice that is used to extract the trade port name and match it to the country while screening. This is done because two countries may not have the same port name, so this avoids duplication. The port name is also matched against the sanctioned port reference data.

### Identifiers

The following identifiers are used for Trade Port matching:

**Table 7-1 Trade Port Identifiers**

| Identifier Description | Standard Prepared Attribute Name |
|---|---|
| Port Name | Port Name |
| Port Code | Port Code |
| Port Synonyms | Port Synonyms |
| Port In Any Order | Port In Any Order |
| Country | Country |

### Clusters

The following clusters are used for Trade Port matching:

- PortName: This cluster uses the Port Name to generate cluster keys. Port Name Cluster allows new records to be matched against existing records in a system.
- PortCode: This cluster uses the Port Code to generate cluster keys. Port Code Cluster allows new records to be matched against existing records in a system.
- PortSynonyms: This cluster uses the Port Synonyms to generate cluster keys. Port Synonyms Cluster allows new records to be matched against existing records in a system.
- PortName In Any Order: This cluster uses the Port Name In Any Order to generate cluster keys. Port Name In Any Order Cluster allows new records to be matched against existing records in a system.
- ISOCountry: This cluster uses the ISO Country to generate cluster keys. ISO Country Cluster allows new records to be matched against existing records in a system.

## 7.1 Trade Port Match Rules

The following match rules are used for Country, city, bad BIC and stop key words advanced match processing:

**Table 7-2 Trade Port Match Rules**

| Rule Code | Matching Rule | Summary of Rule Logic | Example Matching Data Port Name | Example Matching Data Country |
|---|---|---|---|---|
| P002 | Exact port name only | Match against port name column of reference data | Port of Houston | CH |
| P003 | Exact port synonym name and country (ISO) | Match against port synonym column of reference data along with country name where port should belong to that country | Houston Port | VE |
| P004 | Exact port synonym name only | Match against port synonym column of reference data | Houston Port | VE |
| P005 | Exact port code only | Match against port code columns of reference data along with country name where port should belong to that country | 3003 | TW |
| P006 | Port name in any order and country (ISO) | Match against port name in any order against port name column of reference data along with country name where port should belong to that country | ABERDEEN-HOQUIAM, WASH<br>ADDISON USER FEE AIRPORT, DALLA | TW |
| P007 | Port name in any order | Match against port name in any order against port name column of reference data | ABERDEEN-HOQUIAM, WASH<br>ADDISON USER FEE AIRPORT, DALLA | UK |
| P008 | Port name at least one word matching and country (ISO) | Word match count is >0 along with country name where port should belong to that country | ABERDEEN | UK |
| P009 | Port name at least one word matching | Word match count is >0 along with country name | ABERDEEN | UR |
| P010 | Port country name match | Word match count is >0 along with country name where port should belong to that country | ABERDEEN | UK |

**Table 7-2 (Cont.) Trade Port Match Rules**

| Rule Code | Matching Rule | Summary of Rule Logic | Example Matching Data Port Name | Example Matching Data Country |
|---|---|---|---|---|
| P001 | Exact port name and country (ISO) | Match against port name column of reference data along with country name where port should belong to that country | Port of Houston | CH |