

Potential Match Utility

Technical Specification Document

Release Version 1.0



About this Manual

This Technical Specification Document is intended to give you a jump-start working knowledge on Potential Match utility. You will be guided through the features and extract procedures.

Intended Audience

This manual is addressed to Reltio partners and customers.

Overview

The Potential match Extract utility helps to extract the Reltio Matched Entities OV data from the specified tenant by creating a simple configuration properties file. It does not require any programming. In this extraction process, only the values that have "ov=true" along with Match URIs of an attribute come to the extract file.

The Potential Extract Report can be used for match/merge analysis.

Click the following link to access and download the Jar file:

[relio-pm-extractReport-0.0.3.jar](#)

Potential Extract procedure includes the following steps:

1. Match Extract overall steps.
 - a. Step 1: Preparing Attribute Properties File
 - b. Step 2: Preparing job configuration file
 - c. Step 3: Running the Jar to Generate the Extract
2. Output Extract File

Supported Features:

- This tool extracts entity level attributes specified in the attribute properties file that are simple, nested and reference attributes along with matched URIs.
- It supports only CSV file as output.
- It supports Transitive Matches

Extracting Potential Match

The Potential Match extract includes the following steps:

1. Identify the Reltio attributes that need to be extracted to check the Matches and prepare the attribute properties file.
2. Create the Job configuration file which holds output, mapping document path and output file format.
3. Execute the jar file with the created Job configuration properties file.

Step 1: Preparing Attribute Properties File

1. The Attribute Properties file should be either a properties file or a text file .
2. The format of the file is <<Reltio Attribute Name>> = <<No: of values to be extracted>>
3. Number of values to be extracted is optional as the default brings one value.

Sample Attribute Properties File

- FirstName
- LastName
- Address.AddressLine1
- Address.AddressLine2
- Address.City
- Address.State
- Address.Zip5
- Identifiers.Type
- Identifiers.Id

Step 2: Preparing job configuration file

To prepare the job configuration file, you need to declare these properties in the job configuration file:

- RELTIO_SERVER_HOST = <Server Host URL>(Example, <https://sndbx.reltio.com>)
- TENANT_ID = <Tenant Id>
- AUTH_URL = <https://auth.reltio.com/oauth/token>
- ENTITY_TYPE = <HCO or HCP or SoldToCustomer or any other entity type for which you want to generate the extract>
- USERNAME = <Reltio user name of the user>
- PASSWORD = <Reltio password, used to acquire the access token>
- OV_ATTRIBUTE_FILE_LOCATION = <The Name of the attribute properties file with its full path>(Example : C:\\QA_Support_Internal\\OV and XREF Extract Testing\\ov\\Entity_Type_HCO\\product-attributes-ov.properties)
- OUTPUT_FILE_LOCATION = <The Name of the output extract csv file with its full path>(Example : C:\\QA_Support_Internal\\OV and XREF Extract Testing\\ov\\Entity_Type_HCO\\tenant_extract_HCO.csv)
- FILE_FORMAT = <CSV/Flat>

- HEADER_REQUIRED =Yes <If Header row required in the file or not (Yes/No)>
- THREAD_COUNT = <No. of Threads to be used to take the extract > (Example 20)
- BATCH_SIZE=<max entities to extract per api call>
- TRANSITIVE_MATCH=false<Is Transitive matches required or not (true/false)>
- EXTRACT_ALL_VALUES=false <Retrieves the non ov values as well if the flag is true>

Sample Configuration file

```
AUTH_URL=https://auth.reltio.com/oauth/token
USERNAME=mohan.kumar@reltio.com
PASSWORD=
TENANT_ID=9eBTsa2qL8ZgG7e
RELTIO_SERVER_HOST=sndbx.reltio.com
THREAD_COUNT=20(Recommended)
ENTITY_TYPE=HCP
OV_ATTRIBUTE_FILE_LOCATION=hcp-mapping.properties
OUTPUT_FILE_LOCATION=pm-extracts-hcp.csv
FILE_FORMAT=CSV
HEADER_REQUIRED=Yes
BATCH_SIZE=100
TRANSITIVE_MATCH=false(Default)
EXTRACT_ALL_VALUES=false
```

Step 3: Running the Jar to Generate the Extract

Use the following command to run the jar:

```
java -jar <<Jar file Name>>(Example : reltio-pm-extractReport-0.0.3.jar ) <<Job Configuration File Name>>(Example : job_configuration.properties) > <<Name of the log file>>(Example: Tenant_Extract.log)
```

See the following command formats for various operating systems.

Operating System	Command Format
------------------	----------------

Windows	<pre>java -jar reltio-pm-extractReport-0.0.3.jar job_configuration.properties > \$logfilepath\$</pre>
Unix	<pre>nohup java -jar reltio-pm-extractReport-0.0.3.jar job_configuration.properties > \$logfilepath\$ 2>&1 &</pre>

Output Extract File

The output CSV extract file contains the attributes that have been defined in the attribute properties file along with the Matched record URI for all the entities from the tenant. You can see the sample output extract file in the following location:

[TRANSITIVE_pm-extracts-hcp.csv](#)

[NON-TRANSITIVE_pm-extracts-hcp.csv](#)