

Potential Match Util User guide

The Potential match Extract Tool is used to extract the Reltio Matched Entities OV data from the specified tenant by creating a simple configuration properties file without doing any programming.

In this extraction process only those values of an attribute come to the extract file which are having "ov=true" along with Match URIs .

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

The Link to Download the Jar File :

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

- **Supported Features:**
- **Match Extract overall steps.**
 - **Step 1: Preparing Attribute Properties File**
 - **Step 2: Preparing job configuration file**
 - **Step 3: Running the Jar to Generate the Extract**
- **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.**
- **Supports only CSV file as output.**
- **Supports Transitive Matches**

-

Potential Match Extract overall steps.

1. Identify the Reltio Attributes that needs 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

- The Attribute Properties File can be either a properties file or a text file .
- The format of the file is <<Reltio Attribute Name>> = <<No: of values to be extracted>>
- No. 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

Etc...

Step 2: Preparing job configuration file

To prepare the job configuration file we 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 we are generating 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)
- **TRANSITIVE_MATCH**=false<Is Transitive matches required or not (true/false)>
- **BATCH_SIZE**=100 (No of records per call. Default 100)
- **EXTRACT_PER_RULE**=false (Extract the potential matches per rule. If it is true then tool will create multiple files with the matchrule name. Default is false)
- **FILTER_CONDITION**=contains(attributes.LegacyData.ID,'*00') This is an optional parameter and used to filter the records based on filter condition.

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_PER_RULE=true

FILTER_CONDITION=contains(attributes.LegacyData.ID,'*00')

Step 3: Running the Jar to Generate the Extract

To run the jar the following command is used :

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)

On Windows

java -jar reltio-pm-extractReport-0.0.3.jar job_configuration.properties >
\$logfilepath\$

On Unix

nohup java -jar reltio-pm-extractReport-0.0.3.jar
job_configuration.properties > \$logfilepath\$ 2>&1 &

-

Output Extract File

The output CSV extract file contains the attributes which were defined in the attribute properties file along with the Matched record URI for all the entities from tenant .Sample output extract file :

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