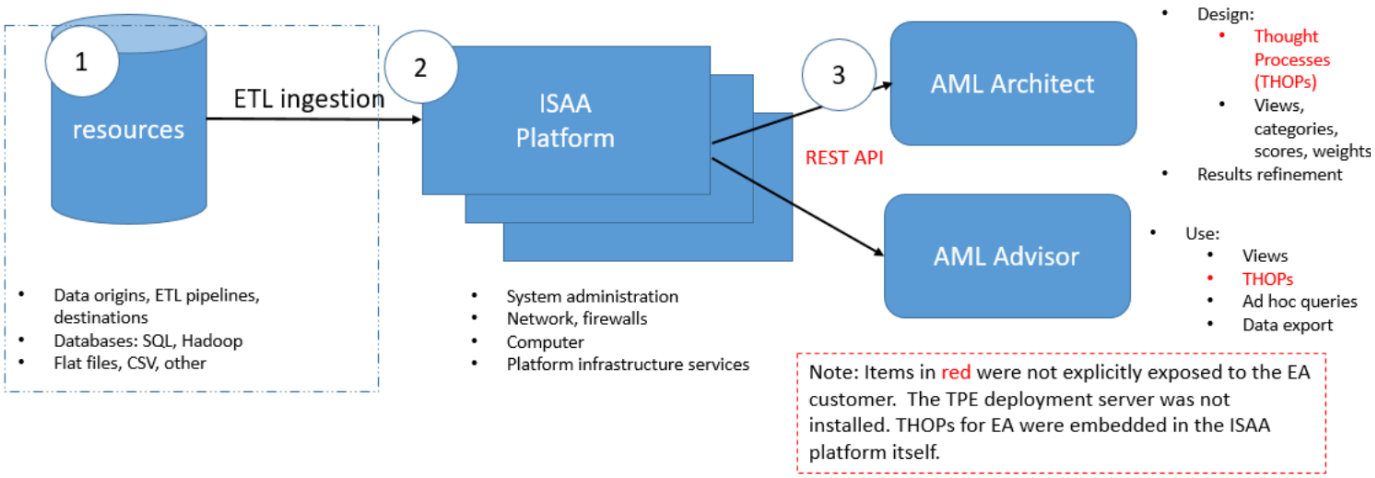
ISAA AML Architect Guide, EA edition

ISAA conceptual overview

The Intel Saffron Anti-Money-Laundering Advisor (ISAA) is a cognitive computing system for financial institutions to discover *actionable insights* in to possible crime. Based on systematic analysis of your data, you can tailor your analyses to your data and your needs to progressively refine analyses and improve insights.

Below is a simplified, at-a-glance logical view of the ISAA and its subsystems.



1	2	3
Your data is central to ISAA. A data source is called a *resource*. AML Architects design *pipelines* (which have an *origin* and a *desti nation*) to run data transformations via an *ETL* (Extract, Transfer, Load) process called *ingestion*. During ingestion, data are normalized, sent to a destination, and made available for queries via the AML Advisor for further refinement and investigation.	The ISAA platform is the central hub of the ISAA system. You configure clusters of *leader nodes* and *worker nodes*. You can also setup *zones* and *spaces* to secure containers, segregate your data, and isolate ISAA processes.	AML investigators work with ISAA's web user interfaces: the AML Architect and the AML Advisor. • With AML Architect, you design specific queries (called *views*) and *THOught Processes* (THOPs) for use via the AML Advisor. Types of views include *anomaly views* and *customer risk views*. • With the AML Advisor, users can also create ad hoc queries with LiveSearch.

Notes on Early Adopter deployment

Throughout these guides, specific details bout the ISAA deployment at a customer site are indicated with this marker:

> Note on EA deployment

In general, hese notes indicate where the EA deployment varied from the ISAA system design, where the ISAA system itself might have been immature, or where additional manual steps had to be taken to successfully deploy.

Early Adopter (EA) audiences and ISAA documentation

This collection of guides describes the Early Adopter (EA) release of ISAA, which was deployed at a customer site.

The ISAA documentation is grouped into usable collections of information by roles (or personas).

In practice at your site, these roles might be combined. For instance, in test/evaluation, these roles are often a single person.

ISAA AML Advisor Guide

The *ISAA AML Advisor Guide* is for "data explorers", persons using the AML Advisor web interface to investigate, query, and analyze results in ISAA, results based on the work of data analysts.

Recommended skills

- Curiosity
- Knowledge of AML
- Understanding of your specific goals for AML

ISAA AML Architect Guide

The *ISAA AML Architect Guide* is for the “data analyst” (sometimes called "data scientist" or "programmer") who designs the Extract, Transform, Load (ETL) programming, ingestion, categories, attributes, application of algorithms via THOught Processes (THOPs), and query design. With the AML Architect web interface, ETL tools, and JavaScript programming, the data analyst acts as “power user” in preparing data for use by data explorers via the AML Advisor.

Recommended skills

- Deep knowledge of your data and the desired goals/result of your design
- Comfort with ETL
- Familiarity or prior experience with machine learning
- Knowledge of computer programming with REST APIs and JavaScript
- Experience with StreamSets helpful

ISAA Administrator Guide, with Installation

The *ISAA Administrator Guide, with Installation* is for system administrators involved installing ISAA and third-party software, configuring servers and services, data resources, network design, maintenance and upgrades, and administration of databases, clusters, and all ISAA components.

Recommended skills

- Comfort with Linux operating system
- Familiarity with TCP/IP networks, firewalls, ports
- Familiarity with software installation using tar, gzip, RPM
- Familiarity with LDAP
- General system administration
- Familiarity with Hadoop-based systems helpful

ISAA Glossary

The ISAA Glossary, EA edition includes definitions of frequent terms in AML and ISAA.

Background: installation and required third-party software

Before you start work with the AML Architect, the ISAA system and required third-party software must be installed. These details are in the ISAA Administrator Guide, with Installation, EA edition.

How the AML Architect works

A high-level statement of how the system works is below.

Phase 1: ETL in front with StreamSets: data *ingestion*:

- You setup *resources*, which are sources of raw data, such as your database.
- In ISAA/StreamSets, you create *pipelines* with *stages,* in which you use *processors* to define the transforms you want to apply to the raw data. This is programming with StreamSets processors, some supplied with ISAA and others you yourself can create.

Phase 2: THOPs on back, after ETL

Note on EA deployment: THOPs were not explicitly exposed to the EA customer.

- Working with the normalized data, you create thought processes (THOPs) by programming algorithms in JavaScript to derive the analyses your users need.
- The thought-processed data is then made available to end-users for exploration via the AML Advisor.

Starting out with your data: discovery

As an AML professional, you know your work, but here are some investigative questions as a generalized approach to beginning data design with ISAA:[1]

- What is the problem?
- Why does the problem need to be solved?
- How would I solve the problem?
- What is the desired AML end goal of your use of ISAA? In plain language, describe the problem-to-solve and why it needs to be solved. Try to describe how you yourself would solve your problem to reach the desired goal.

[1] Dr. Jason Brownlee, machine learning practitioner, at https://machinelearningmastery.com/how-to-define-your-machine-learning-problem/

Flow of ETL ingestion pipelines in StreamSets Data Collector

The process of a *pipeline* follows this general flow. "Pipeline" is a term specific to STreamSets. The more general term for this process is *ingestion*.

A pipeline consists of:

1. Connections to the *resource* (data source), which depends on the type of resource. This is called the *origin*.
2. Successive *stages* in the pipeline for processing the input data from the resource. In each stage you define a *processor* that transforms the data to meet your needs, such as extracting attributes, normalizing dates, and other functions, including error handling. The results of the stage can be piped directly to the next stage or can be routed to other stages via branching logic.

   In addition to processors from StreamSets, processors supplied with ISAA are described in ISAA AML Architect processors for StreamSets pipelines and the remainder of this guide.
3. Connections to *destinations* to store the results of the pipeline.

Example of a simple pipeline

This is an example of a simple, two-stage pipeline with processors supplied with ISAA.

> This example does not include defining the origin or the destination of the pipeline.

1. In one stage, you can use the Sentence Segmenter processor to pull out entire sentences from a text field.
2. Then, you can run the results of the first stage through a second stage with the Sentence Splitter processor to break the text from the first stage into more usable parts, such as eliminating the non-human readable parts of a log file record.

Pre-installing and configuring StreamsSets Data Collector parcel in Cloudera Manager

The ISAA documentation does not describe your exact StreamSets configuration, your resources, origins, or destinations. Installing the StreamSets Data collector as a parcel in Cloudera Manager is discussed in the ISAA Administrator Guide, with Installation, EA edition. Consult with your system administrators on the exact configuration.

Essential doc: StreamSets Data Collector User Guide and more

ISAA documentation does not include step-by-step details for creating and managing pipelines.

You should become familiar with the documentation provided by StreamSets:

- The StreamSets Data Collector User Guide at https://streamsets.com/documentation/datacollector/latest/help/#Getting_Started/GettingStarted_Title.html#concept_htw_ghg_jq
- Getting started at https://streamsets.com/documentation/datacollector/latest/help/#Getting_Started/GettingStarted_Title.html#concept_htw_ghg_jq
- A tutorial at https://streamsets.com/documentation/datacollector/latest/help/#Tutorial/ExtendedTutorial.html.

ISAA AML Architect processors for StreamSets pipelines

This is reference information about the ISAA AML Architect processors for use in ETL ingestion pipelines in StreamSets Data Collector.

> You need to define only those processors that best suit your data, not all of them.

Transform	Description
Attribute Filter	Filters and merges extracted attributes.
Document Parser	Extracts text and metadata from binary data in MS Office, PDF documents.
Field Mover	Moves fields in a StreamSets record to a different location. Use the Field Mover when you want to create new Saffron resource segments within a record.
Geocode Lookup	Based on a provided country code and zip code, attempts to look up geographic location information (geocode) for a global mailing address.
HTML Tag Filter	Removes HTML tags from a StreamSets text field and converts the input to plain text.
Namelist Extractor	Extracts a set of Saffron attributes from one or more record text fields. The processor uses the concept of a *namelist* to normalize variants of domain specific values.
Namelist Translator	Creates new record fields and data from a StreamSets text field. The processor uses the concept of a *namelist* to normalize variants of domain specific values.
Regex Extractor	Extracts a set of Saffron attributes from one or more StreamSets text fields using the regular expressions.
Regex Translator	Extracts values from input fields based on provided regular expressions and creates a field containing a list of those values. This extractor can also be used to create new Saffron resource segments within a record and add the extracted values into the new segments.
Regex Field Extractor	Uses regular expressions to match and translate StreamSets text fields. Depending on how the processor is configured, the original text might not be preserved.
Saffron Memory Base Resource Generator	Generates Saffron Memory Base resources in JSON format for upload.
Saffron Memory Base Uploader	Uploads a resource generated by the Saffron Memory Base Resource Generator to a Saffron Memory Base system using the REST API.
Stanford Entity Extractor	Analyzes plain text to derive the base forms of words and their parts of speech (whether they are names of companies, people, and so on). It also normalizes dates, times, and numeric quantities. It marks up the structure of sentences into phrases and word dependencies. Certain parts of speech can be extracted as attributes, for example, noun and verbs.
Sentence Segmenter	Extracts sentences from a specified StreamSets text field.
Sentence Splitter	Splits large sentences in StreamSets text fields. Making sentences shorter can make resource-intensive processing steps like *NLP* faster. Typically, this processor should be used to split long running sentences that might not include human readable text, for example, a log file.
Temporal Elaborator	Extracts Saffron date and time attributes from a Streamsets Date, Time or DateTime field.
WordNet Extractor	Extracts the different parts of speech (such noun, verb, adjective) from a StreamSets text field.

Configuring processors in the StreamSets Data Collector

You define the processors provided by ISAA in pipeline stages in the StreamSets Data Collector:

- The steps in the following sections assume that you are already logged into the Data Collector.
- For each stage in your pipeline, to define the processor, you cycle through the tabs in the Data Collector to enter definitions in the indicated properties. These tabs are identified below: General, Configuration, Advanced and others tabs specific to the processor.

Careful with parallel processing of records

Many of the processor definitions have an **Advanced** property called **Process Records in Parallel**.

In general, you should use this option with caution, because it can adversely impact system performance.

Attribute Filter

General tab

Attributes Filter General Property	Description
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all preconditions are processed based on the error handling configured for the stage. Click **Add** to create more preconditions.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.
Required Fields	Fields that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all required fields are processed based on the error handling configured for the pipeline.
Description	Optional description.
Name	Stage name.

Configuration tab

Attributes Filter Configuration Property	Description
Input Fields	Fields to process
Categories To Remove	List of regular expressions identifying the categories to be removed after filtering.

Attribute overlap tab

Attributes Filter Attribute Overlap Property	Description
Processor Overlaps	Specify processors that allow overlaps (optional).

Processor	Processor attribute results that are allowed to overlap. Use the **Stage Instance Name** from the **Info** tab of the processor in question.

Advanced tab

Attributes Filter Advanced Property	Description
Process Records in Parallel	The processor attempts to process records in each batch in parallel. Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads used to process records in a single batch.

Document Parser

The Document Parser extracts text and metadata from binary data in MS Office and PDF documents. Each input field must be a byte array containing the binary contents of the file. Streams fitting this purpose can be produced by any StreamSets origin that supports the binary data type.

Reading binary files with JavaScript

The **Directory** origin does not support direct reading binary files. To read the required files, configure the **Directory** origin data type **Whole File**. Immediately, following the **Directory** origin, add a JavaScript evaluator to read the contents of the file into a byte array. In the **Script** section of the JavaScript evaluator, add the following JavaScript.

Depending on the size of the input files, the size of the byte array might need to be increased or decreased in the declaration `var data = new ByteArray(1024000)` shown below.

```
for(var i = 0; i < records.length; i++) {
  try {
    var is = records[i].value['fileRef'].getInputStream();
    var ByteArrayOutputStream =
  Java.type("java.io.ByteArrayOutputStream");
    var buffer = new ByteArrayOutputStream();
    var ByteArray = Java.type("byte[]");
    var data = new ByteArray(1024000);
    while ((nRead = is.read(data, 0, data.length)) != -1) {
        buffer.write(data, 0, nRead);
    }
    var newRecord =
  sdcFunctions.createRecord(records[i].sourceId);
    newRecord.value = buffer.toByteArray();
    buffer.flush();
    is.close();
    output.write(newRecord);
  } catch (e) {
    error.write(records[i], e);
  }
}
```

General tab

Document Parser General Property	Description
Name	Stage name.
Description	Optional description.
Required Fields	Fields that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all required fields are processed based on the error handling configured for the pipeline.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create more preconditions.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.

Input/Output Config tab

Document Parser Input/Output Config Property	Description

Destination Naming Strategy	Strategy for naming destination attribute names. • Same Field - The filtered text is placed in the same field as the corresponding input field. • Prefix / Suffix - The filtered text is placed in a new field using the **Destination Field Prefix** and **Destination Field Suffix** configuration properties to define the new field name. • Explicit - The filtered text is placed in a new field. The Input/Output fields are defined in the Input/Output configuration property.
Original Field	Action for the original fields. • Keep - The original field is retained. • Remove - The original field is removed.
Input Fields	List of fields to process.
Extract Metadata?	Metadata for document is extracted. Metadata includes the author name, company, and other fields specified in the document properties.

Advanced tab

Document Parser Advanced Property	Description
Process Records in Parallel	The processor attempts to process records in each batch in parallel. Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads used to process records in a single batch.

Field Mover

The Field Mover moves fields in a StreamSets record to a different location. Use the Field Mover when you want to create new Saffron resource segments within a record.

There are two general use cases for this processor:

- To move an existing field to an existing Map field within the record.
- To create a new Map field and move one or more fields into the newly created Map.

General tab

Field Mover General Property	Description
Name	Stage name.
Description	Optional description.
Required Fields	Field that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.

On Record Error	Error record handling for the stage. • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.

Configuration tab

Field Mover Configuration Property	Description
Destinations	To move fields to new Destination, enter or select the **Destination** field, and then select one or more **Input Fields** fr om the existing record. If the Destination field does not exist, a new Map is created and all Input Fields are moved to that Map. If the Destination already exists, the Input Fields are added to the existing field. Click **Add** to move additional fields.

Advanced tab

Advanced Property	Description
Process Records in Parallel	The processor attempts to process records in each batch in parallel. Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads to process records in a single batch.

Geocode Lookup

Geocode Lookup attempts to pinpoint the longitude and latitude of a location on the surface of the Earth (the geocode). This processor relies country and city or zip code information that you supply to approximate a geocode location. Additional information such as country specific region information can be used to further refine the result.

> The physical street address is not used to ascertain the geocode.

The processor expects that all mailing address information is held in a StreamSets MAP or LIST-MAP structure. Below are some examples of how the processor resolves data:

Resolve by country and postal code

In this case the country and postal code information is used to return a geocode location of 35.324, -106.7099.

```
{
      "country": "US",
      "postalCode": "87144"
}
```

Resolve by country and city

In this case the country and city information is used to return a geocode location of 40.7143, -74.006.

```
{
      "country": "US",
      "city": "New York City"
}
```

Resolve when postal code information not available

In this case, the country and postal code is valid, but we do not have postal code information for South Korea. Instead, the processor uses the country/city combination to find the geocode location of 37.566, 126.9784.

```
{
    "country": "KR",
    "city": "Seoul",
    "postalCode": "137-857"
}
```

Resolve when country and city are ambiguous

In this case, the country and city are valid, but there are multiple cities with the same name in the country. For instance, there are at least two cities in the USA named “Portland”: Portland, Oregon (OR) and Portland, Maine (ME).

The processor attempts to resolve by selecting the city with the largest population with a geocode location of 45.5234, -122.6762.

```
{
      "country": "US",
      "city": "Portland"
}
```

Compare this input data to the next example, in which the political state refines the result.

Resolve country and city with additional country-specific regional information

If the country and city are valid, the regional information (political state or province) can be used to further refine the search, reducing the possibility of duplicate city matches. In this case the geocode location is 43.6615, -70.2553.

```
{
      "country": "US",
      "city": "Portland",
      "state": "ME"
}
```

General tab

Geocode Lookup General Property	Description
Name	Stage name.

Description	Optional description.
Required Fields	Field that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.

Configuration tab

Geocode Lookup Configuration Property	Description
Number of decimal places	Specify the number of decimal places to use for the geocode location. Default: Four decimal places. This is also the maximum number of decimal places
Error if not found?	If checked, when a geocode is not found for a specified addresss, the record is treated as an error record subject to the error handling defined for the pipeline. If not checked, record processing continues and no geocode information for that address is added to the record.

Address Fields tab

Configure the following properties for each **Address Information** entry.

Address Fields General Property	Description
Root field path	The StreamSets field that contains all the address information. This field should be a StreamSets MAP or LIST-MAP field type, although you can specify wildcards for an array of MAP or LIST-MAP objects.
Country field	Name of the field within the MAP or LIST-MAP that contains the field that is the ISO 3166-1 alpha-2 letter country abbreviation.
Region field	Name of the field within the MAP or LIST-MAP that contains a country specific region code that can optionally be used to further refine a city search. For example, in the United States this would be a state, but in Canada, this would be a province.
City field	Name of the field within the MAP or LIST-MAP that contains city information.
Postal code field	Name of the field within the MAP or LIST-MAP that contains postal code information.
Output field	Location within the MAP or MAP-LIST to place the geocode information found for the address.

Advanced tab

Address Fields Advanced Property	Description

Process Records in Parallel	The processor attempts to process records in each batch in parallel. Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads used to process records in a single batch.

HTML Tag Filter

The HTML Tag Filter removes HTML tags from a StreamSets text field, converting the input to plain text.

General tab

General Property	Description
Name	Stage name.
Description	Optional description.
Required Fields	Field that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.

Input/Output Config tab

Input/Output Config Property	Description
Destination Naming Strategy	Strategy for naming destination attribute names. • Same Field - The filtered text is placed in the same field as the corresponding input field. • Prefix / Suffix - The filtered text is placed in a new field using the Destination Field Prefix and Destination Field Suffix configuration properties to define the new field name. • Explicit - The filtered text is placed in a new field. The Input/Output fields are defined in the Input/Output configuration property.
Original Field	Action for the original fields. • Keep - The original field is retained. • Remove - The original field is removed.
Input Fields	List of fields to process.
Destination Field Prefix	Prefix to be applied to the destination attribute name.
Destination Field Suffix	Suffix to be applied to the destination attribute name.

Input/Output	List of input/output parameters: • Input Field - Input field to process. • Output Field - Output field for filtered text.

Advanced tab

Advanced Property	Description
Process Records in Parallel	The processor attempts to process records in each batch in parallel. Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads used to process records in a single batch.

Namelist Extractor

The Namelist Extractor extracts a set of Saffron attributes from one or more record text fields. The processor uses the concept of a *namelist* to normalize variants of domain-specific values.

See also the related processor Namelist Translator.

Definition of a namelist

A namelist entry consists of the following components. These are variables that you supply for your specific definition.

- *category* - Saffron attribute category name that is used for any extracted attributes.
- *canonical* - The Saffron attribute value that is used for any extracted attributes.
- *variants* - This is the optional set of variant values that is mapped to the canonical value if found in the record field.

Namelist example

Given a namelist entry with the following values...

Entry	Value
category	city
canonical	New York City
variants	NYC, The Big Apple

And also given the following sentence for processing:

```
I bought airline tickets to NYC, I will be in The Big Apple for New Year's Eve.
```

Result: The following two Saffron attributes would be extracted from the sentence above:

Category	Original value that triggered extraction	Resulting value
city	`NYC`	`New York City`
city	`The Big Apple`	`New York City`

General tab

Namelist Extractor General Property	Description

Name	Stage name.
Description	Optional description.
Required Fields	Field that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.

Configuration tab

Namelist Extractor Configuration Property	Description
Input Fields	Fields to extract attributes from.
Case Sensitive	If checked, all namelist comparisons for matches are case-sensitive.
Allow underscore as word boundary	If checked, an underscore character is used as a valid word boundary when making namelist comparison matches.
Use hierarchical category naming	If checked, all found attribute categories are prefixed or suffixed based on the **Hierarchical name location** by their field name using the separator provided in the **Hierarchical naming separator**. See Example of property “Use hierarchical category naming”.
Hierarchical name location	Specifies where to add the hierarchical naming information. • Place before category name - Places the field name as a prefix to the attribute category with the **Hierarchical naming separator** as a separator between the field name and the attribute category. • Place after category name - Places the field name as a suffix to the attribute category with the **Hierarchical naming separator** as a separator between the field name and the attribute category.
Hierarchical naming separator	Separator between the field name and the attribute category. See Example of property “Use hierarchical category naming”.

Example of property “Use hierarchical category naming”

This property constructs a hierarchical organization of categories as Saffron attributes.

Property settings	Input field and input categories	Resulting hierarchy
Hierarchical name location set to **Place before category name**		
Hierarchical naming separator set to period (".")		
	`title` field and `noun` categories	
		`title.noun`

CSV tab

Namelist Extractor CSV Property	Description
CSV Namelist	A set of namelist entries to convert. Each line has the format: *category, canonical, variant_1, variant_2, ..., variant_N*

Advanced tab

Namelist Extractor Advanced Property	Description
Process Records in Parallel	The processor attempts to process records in each batch in parallel. Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads used to process records in a single batch.

Namelist Translator

The Namelist Translator creates new record fields and data from a StreamSets text field. The processor uses the concept of a *namelist* to normalize variants of domain-specific values.

See related property Namelist Extractor.

The Namelist Translator has two different purposes:

- Create new fields.
- Remove existing fields.

Definition of a namelist

See Definition of a namelist.

Creating new fields

Items matched via the namelist are extracted into new StreamSets text fields.

The location of the new fields depends on how you configure the processor.

Namelist example

Given a namelist entry with the following values...

Entry	Value
category	`City`
canonical	`New York City`
variants	`NYC, The Big Apple`

And given the following sentence to process...

```
I bought airline tickets to NYC, I will be in The Big Apple for New Year's Eve.
```

Result: The following two Saffron attributes would be extracted from that original sentence.

Category	Original value that triggered extraction	Resulting value

city	NYC	New York City
city	The Big Apple	New York City

Removing a field

A different configuration for this processor is to remove a field that contains any of the variants. If a variant is found in text field, that field is removed from the StreamSets record.

General tab

Namelist Translator General Property	Description
Name	Stage name.
Description	Optional description.
Required Fields	Field that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.

Configuration tab

Namelist Translator Configuration Property	Description
Input Fields	Fields to translate
On Match	How to handle matches: • Create new Attributes from match Namelist values - Creates a new field based on the category value, containing the canonical value for each match of any variant field found in the text. • Remove matches attributes from record - Removes the field if it finds a match of a variant.
Case Sensitive	If checked, use case-sensitive rules when matching tokens for translation.
Original Field	If the **On Match** Create new Attribute from match Namelist values option is selected, you can select an action for the original fields. • Keep - The original field is retained. • Remove - The original field is removed.
Preserve attribute name?	If checked, all found attributes are placed in the original field name, which may also contain the original field, depending on the **Original Field** setting. Otherwise, a new field is created based on the category name for the found variants.

CSV tab

Namelist Translator CSV Property	Description
CSV Namelist	A set of namelist entries to convert. Each line has the format: *category, canonical, variant_1, variant_2, ..., variant_N*

Advanced tab

Namelist Translator Advanced Property	Description
Process Records in Parallel	The processor attempts to process records in each batch in parallel. Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads used to process records in a single batch.

Regex Extractor

The Regex Extractor extracts a set of Saffron attributes from one or more StreamSets text fields using the regular expressions.

See related processor Regex Translator.

Regex Extractor example

The reult we want is to extract the monetary unit and amount from the following input text:

```
$100
```

Below are the regular expressions to extract the desired portions of the string.

Category	Regexp	Result of extraction
monetary_unit	`\W`	`$`
amount	`\w+`	`100`

General tab

Regex Extractor General Property	Description
Name	Stage name.
Description	Optional description.
Required Fields	Field that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.

On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.

Configuration tab

Regex Extractor Configuration Property	Description
Input Fields	Fields to process.
Parameters	List of Category and Regular Expression pairs. **Category** - Category name of the new attribute. **Regular Expression** - Regular expression to match for the corresponding category.
Use hierarchical category naming	If checked, all found attribute categories are prefixed or suffixed based on the **Hierarchical name location** by their field name using the separator specified in the **Hierarchical naming separator**. See example below.
Hierarchical name location	Specifies where to add the hierarchical naming information. • Place before category name - Places the field name as a prefix to the attribute category with the **Hierarchical naming separator** as a separator between the field name and the attribute category. • Place after category name - Places the field name as a suffix to the attribute category with the **Hierarchical naming separator** as a separator between the field name and the attribute category.
Hierarchical naming separator	Separator between the field name and the attribute category.

Example of “Use hierarchical category naming”

For example, if **Hierarchical name location** were set to **Place before category name** and **Hierarchical naming separator** was set to a period ("."), all noun categories found in a field called title would result in Saffron attributes with a category name of title.noun.

Advanced tab

Regex Extractor Configuration Property	Description
Process Records in Parallel	The processor attempts to process records in each batch in parallel. Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads used to process records in a single batch.

Regex Translator

The Regex Translator uses regular expressions to match and translate StreamSets text fields. Depending on how you define the processor, you can preserve the original text or discard it.

As an example, consider the situation when the format of a part number must be translated from one format to another. The format of the old part number is three capital letters followed by three numbers. It must be translated to a new format in which the letters and numbers are separated with an underscore.

The regex and “To Value” fields are described in Configuration tab below.

Input string	Regex	“To Value” parameter	Result
`Part number ABC123 was fixed`	`([A-Z]{3})([0-9]{3})`	`$1_$2`	`Part number ABC_123 was fixed.`

General tab

Regex Translator General Property	Description
Name	Stage name.
Description	Optional description.
Required Fields	Field that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.

Configuration tab

Regex Translator Configuration Property	Description
Input Fields	Fields to process
Perform all translations?	If checked, all translations apply. Otherwise, processor stops after the first match is found.
Original Field	Action for the original fields. • Keep - The original field is retained. The input text field is converted into a List and the original text is the first element of the list. The translated text is the second element of the list. • Remove - The original field is removed and replaced with the translated text.
Translations	List of regex translations: • From Regex Pattern - Regex used to find translation items • To Value - Value to replace regex matches with. If the From Regex Pattern contains groups, the value matched in the group may be referenced using $N where N is the number of the group. See also Regex Extractor example.

Advanced tab

Regex Translator Configuration Property	Description

Process Records in Parallel	The processor attempts to process records in each batch in parallel. Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads used to process records in a single batch.

Regex Field Extractor

The Regex Field Extractor extracts values from the input fields based on the provided regular expression and creates a field containing the list of those values. The Regex Field Extractor can also be used to create new Saffron resource segments within a record and add the extracted values into the new segments.

Example of Regex Field Extractor

This example illustrates placing the extracted fields in either the same segment or a new segment.

Assume the property settings for the Regex Field Extractor in the first column of the table below.

Also assume the following input record:

```
City, Portland, State, Oregon
```

Results: Running the Regex Field Extractor processor against the record produces the following results:

Example of Regex Extractor Input/Output Config Properties	Result

Destination Naming Strategy: Same Field
Value Regex: `\w+`
Input Fields: `/text`
Segmentation: Place results in the same segment as the source attribute
	`/address (List[5])`
	`- City, Portland, State, Oregon`
	`- City`
	`- Portland`
	`- State`
	`- Oregon`

Destination Naming Strategy: Regular expression
Value Regex: \w+
Output Field Regex: \w+
Input Fields: `/text`
Segmentation: Place results in a single new segment
Segment Location: `/output`
	`/output (Map)`
	`-/City (List[1])`
	`- Portland`
	`-/State (List[1])`
	`- Oregon`

General tab

Regex Extractor General Property	Description
Name	Stage name.
Description	Optional description.

Required Fields	Fields that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.

Input/Output Config tab

Regex Extractor Input/Output Property	Description
Destination Naming Strategy	Strategy for naming destination attribute names. • Same Field - Adds the list of regex extracted values into the input field. • Prefix / Suffix - Creates a field by prefixing and/or suffixing the input field name with the prefix and/or suffix specified by the user. This field contains the list of regex extracted values. • Explicit - Creates a field with the user provided output field name and contains the list of regex extracted values. • Regular Expression - Creates the output field name using the regular expression provided for output field and extracts the value based on the regex for value. It continues to generate the output field and value alternatively until the end of the content of input field.
Original Field	Select an action for the original field: • Keep - The original field is retained. • Remove - The original field is removed.
Value Regex	Regular expression used to identify field values.
Input Fields	Fields to process.
Destination Field Prefix	If the **Destination Naming Strategy** Prefix/Suffix option is selected, specify the prefix to be applied to the destination attribute name. Either Prefix or Suffix or both can be specified.
Destination Field Suffix	If the **Destination Naming Strategy** Prefix/Suffix option is selected, specify the suffix to be applied to the destination attribute name. Either Prefix or Suffix or both can be specified.
Input/Output	If the **Destination Naming Strategy** Explicit option is selected, specify the list of Input field names and their corresponding output field names. • Input Field - Field to process. • Output Field - Output field to be created.
Output Field Regex	If the **Destination Naming Strategy** Regular Expression option is selected, provide the regular expression to find the output field name.

Segmentation tab

Regex Extractor Segmentation Property	Description

Segmentation Policy	Segmentation Policy to be used when creating new fields. • Place results in the same segment as the source attribute - Places the new field in the same segment as the **Input fields** of **Input/Output** Config tab. • Place results in a single new segment - Places the new field in a single new segment whose location is specified in **Segment Location.** • Place results in separate new segments - Places the new field with each of its value in separate segments in the location specified by **Segment Location.**
Segment Location	Location of the segmented results.

Advanced tab

Regex Advanced Property	Description
Process Records in Parallel	The processor attempts to process records in each batch in parallel. Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads used to process records in a single batch.

Saffron Memory Base Resource Generator

The Saffron Memory Base Resource Generator generates Saffron Memor yBase resources in JSON format for upload.

General tab

SMB Resource Generator General Property	Description
Name	Stage name.
Description	Optional description.
Required Fields	Field that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.

Configuration tab

SMB Resource Generator Configuration Property	Description
Resource Path	The path to which the generated resource will be written. An error is generated if the record already has a value in this field.
Resource ID Expression	Expression for assigning a unique id for each record.
Resource ID Path	The path to which the generated resource ID will be written. An error is generated if the record already has a value in this field.
Delete source data	If checked, deletes the original source data in the generated resource.
Flatten source hierarchy	If checked, flattens the resource by moving all the attributes to the root level.
Create error record if value length too long	If checked, records containing values with length greater than 255 are treated as error records.

Saffron Memory Base Uploader

The Saffron Memory Base Uploader uploads a resource generated by the Saffron Memory Base Resource Generator to a Saffron Memory Base system with the `/uploads` REST API.

NOTE on EA deployment: The `/uploads` API sends data to the TPE deployment server, which was not installed for the Early Adopter program.

General tab

SMB Uploader General Property	Description
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.
Required Fields	Fields that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
Description	Optional description.
Name	Stage name.

Connection tab

SMB Uploader Connection Property	Description
URL	URL of the Saffron Memory Base system.
Zone ID	Zone ID of the zone to which the resource will be uploaded.
Space ID	Space ID of the space to which the resource will be uploaded.

Resource tab

SMB Uploader Resource Property	Description
Resource path	The path in the record to the resource that has to be uploaded.
Resource ID path	The path in the record to the resource ID.

Credentials tab

SMB Uploader Credentials Property	Description
Username	The Saffron Memory Base username for uploading the resource.
Password	The Saffron Memory Base password for uploading the resource.

HTTP tab

SMB Uploader HTTP Property	Description
Maximum Request Time (seconds)	Maximum time to wait for each request completion.
Maximum Total Request Time (sec)	Maximum time to wait for all requests to complete.
Connect Timeout (milliseconds)	HTTP connection timeout. 0 = no timeout.
Read Timeout (milliseconds)	HTTP read timeout. 0 = no timeout.
Maximum Parallel Requests	Maximum number of requests to run in parallel. Use this option with caution, because it can adversely impact system performance.
Use Proxy	If checked, a proxy is used to connect to the URL, which you specify on the Proxy tab.

Proxy tab

SMB Uploader Proxy Property	Description
Proxy URL	The proxy URL.
Username	The proxy username.
Password	The proxy password.

SSL/TLS tab

SMB Uploader SSL/TLS Property	Description
Path to Trust Store	The path to the truststore, complete with file name.
Password	The truststore password.
Path to Key Store	The path to the keystore, complete with file name.

Password	The keystore password.

Stanford Entity Extractor

The Stanford Natural Language Processor (NLP) technology analyzes plain text and provides the base forms of words and their parts of speech (whether they are names of companies, people, etc.). It also normalizes dates, times, and numeric quantities. It marks up the structure of sentences in terms of phrases and word dependencies. Certain parts of speech can be extracted as attributes, e.g. noun.

There are two general uses for this processor:

- To tokenize the input text.
- More sophisticated processing including *lemmatization*, parts of speech identification, and Named Entity Recognition (*NER*).

General tab

Stanford Entity Extractor General Property	Description
Name	Stage name.
Description	Optional description.
Required Fields	Field that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.

Configuration tab

Stanford Entity Extractor Configuration Property	Description
Input Fields	Fields that the Stanford Entity Extractor should process
Category Filter type	Choose how the categories should be filtered
Lowercase text	If checked, input text is converted to lowercase before the attribute extraction begins.
Use hierarchical category naming	If checked, all found attribute categories are be prefixed or suffixed based on the **Hierarchical name location** b y their field name using the separator provided in the **Hierarchical naming separator**. See Example of property "Use hierarchical category naming".
Hierarchical name location	Specifies where to add the hierarchical naming information. • Place before category name - Places the field name as a prefix to the attribute category with the **Hierarchi cal naming separator** as a separator between the field name and the attribute category. • Place after category name - Places the field name as a suffix to the attribute category with the **Hierarchical naming separator** as a separator between the field name and the attribute category.

Hierarchical naming separator	Separator between the field name and the attribute category.

Entity Selection tab

Entity Selection Property	Description
Entity Types	Entity types to extract. For example, date, duration, location, misc, money, number, ordinal, organization, percent, person, set, or time.
Parts Of Speech Handling	Choose whether to extract parts of speech as entities or not.
Generalize Parts of Speech	Choose to condense parts of speech to more general parts of speech. For example, the base verb is used instead of a verb's past participle.
Extract entity as lemma	Use lemma form of an entity. For example, forces is extracted as force.

Blacklisting and whitelisting categories

To configure the **Blacklist** tab or **Whitelist** tab, on the Configuration tab, select blacklist in the **Category Filter Type**. Then, configure the following properties:

Blacklist/Whitelist Property	Description
Blacklist	Specify categories that should be excluded from the output. For example, if noun.artifact is specified, this category is excluded in the output.
Whitelist	Specify categories here that should be included in the output. For example, if noun.artifact is specified, this category is included in output.

Skip Word Lists tab

Skip Word Lists Property	Description
Use Default Skip Word List	If checked, uses the default skip word list to skip unimportant or invalid tokens created during NLP. Default Skip Word List consists of frequently encountered tokens, to skip while processing. If unchecked, you need to provide a custom skip word list.
Custom Skip Word List	Define your own case-insensitive tokens to skip.
Case-sensitive skip word list	Define case-sensitive tokens to skip.

Advanced tab

Stanford Entity Extractor Advanced Property	Description
Process Records in Parallel	The processor attempts to process records in each batch in parallel. **Note:** Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads to process records in a single batch.
Disable	Check this to disable parallel processing.

Processing timeout	This is the maximum number of seconds to wait for processing to complete, before generating an error. 0 = no timeout.
Stanford NLP Properties	Properties to initialize the Stanford NLP. • **Name** - Enter the property name. For example: **annotators**, **tokenize**, **whitespace**. • **Value** - Enter the property value. For example: **tokenize**, **ssplit**, **pos**, **lemma**, **ner**. The default behavior is lemmatization, parts of speech identification, and Named Entity Recognition (NER). To only tokenize the input text, add the **annotators** property with the value **tokenize**.

Sentence Segmenter

The Sentence Segmenter extracts sentences from a specified StreamSets text field.

General tab

Sentence Segmenter General Property	Description
Name	Stage name.
Description	Optional description.
Required Fields	Field that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.

Configuration tab

Sentence Segmenter Configuration Property	Description
Input Fields	Fields to process.
Valid Abbreviations	A list of valid abbreviations that use a period ('.') in their names that must not be confused with the end of a sentence.

Maximum Attributes per sentence	Defines maximum number of extracted attributes in a sentence. Sentences containing attributes greater than this number is split. 0 = no limit on the number of attributes per segment. This is a protection mechanism to guard against mis-segmentation, which could result in many associations.

Advanced tab

Sentence Segmenter Advanced Property	Description
Process Records in Parallel	The processor attempts to process records in each batch in parallel. Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads used to process records in a single batch.

Sentence Splitter

The Sentence Splitter splits large sentences in StreamSets text fields. Shortening sentences can make expensive processing steps like NLP faster. Typically, this processor should be used to split long running sentences that may not be human readable, for example, a log file records.

General tab

Sentence Splitter General Property	Description
Name	Stage name.
Description	Optional description.
Required Fields	Field that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.

Configuration tab

Sentence Segmenter Configuration Property	Description
Input Fields	Fields to process

Maximum Sentence Size	Maximum number of characters for each sentence. Must be 25 characters or greater. • If a sentence exceeds the maximum number of characters, a period (“.”) is inserted to split the sentence. • If this property is not specified, sentences are not modified.
Delete long sentences	If checked, sentences longer than the **Maximum Sentence Size** are deleted.
Truncate Long Sentences	If checked, sentences longer than **Maximum Sentence Size** are truncated. If not checked, the sentence delimiter specified in the **New Sentence Delimiter** property is used to break up the sentence.
New Sentence Delimiter	If creating artificial sentences to conform to the maximum sentence length, the character to insert. If you are *not* deleting long sentences via the **Delete long sentences** property, you must set a delimiter.

Advanced tab

Sentence Segmenter Advanced Property	Description
Process Records in Parallel	The processor attempts to process records in each batch in parallel. Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads used to process records in a single batch.

Temporal Elaborator

The Temporal Elaborator extracts Saffron date and time attributes from a StreamSet’s Date, Time or DateTime field.

General tab

Temporal Extractor General Property	Description
Name	Stage name.
Description	Optional description.
Required Fields	Field that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.

On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.

Configuration tab

Temporal Extractor Configuration Property	Description
Input Fields	Fields to process.
Temporal elaboration types	Types of date and time Saffron attributes to extract from Date or DateTime data types: • YEAR - Creates a year attribute from a Date or DateTime field. • QUARTER_OF_YEAR - Creates an attribute representing the calendar quarter (Q1, Q2, Q3, or Q4) that includes the specific date. • MONTH_OF_YEAR - Creates an attribute representing the month. • MONTH_OF_QUARTER - Creates an attribute representing the year. • MONTH - Creates an attribute of the format *yyyy-mm* representing the year and month. • WEEK_OF_YEAR - Creates an attribute representing the week number of the year. • WEEK_OF_QUARTER - Creates an attribute representing the week number in the calendar quarter. • WEEK_OF_MONTH - Creates an attribute representing the week number in the month. • WEEK - Creates an attribute of the format *yyyy* w*mm* (for example, 1980 w01) representing the week. • DAY_OF_YEAR - Creates an attribute of the day count of year. • DAY_OF_QUARTER - Creates an attribute of the day count in the calendar quarter. • DAY_OF_MONTH - Creates an attribute of the day count in the month. • DAY_OF_WEEK - Creates an attribute of the day in the. • DAY - Creates an attribute of the format *yyyy-mm -dd* (for example, 1980-01-23) representing the day. The following times work on a DateTime field: • HOUR_OF_DAY - Creates an attribute of the hour. • HOUR - Creates an attribute of the format *yyyy-mm-dd hh* representing the year, month, day, and hour. • MINUTE_OF_HOUR - Creates an attribute of the minute. • MINUTE - Creates an attribute of the format *yyyy-mm-dd hh*:*mm* representing the year, month day, hour, and minute. • SECOND_OF_MINUTE - Creates an attribute of the second. • SECOND - Creates an attribute of the format *yyyy-mm-dd hh*:*mm:ss* representing the year, month, day, hour, minute, and second.

Advanced tab

Advanced Property	Description
Process Records in Parallel	The processor attempts to process records in each batch in parallel. Use this option with caution, because it can adversely impact system performance.
Number of Threads	If **Process Records in Parallel** is checked, this property specifies the number of threads used to process records in a single batch.

WordNet Extractor

The WordNet Extractor extracts the different parts of speech (such as noun, verb, adjective) from a StreamSets text field.

General tab

WordNet Extractor General Property	Description
Preconditions	Conditions that must evaluate to TRUE to allow a record to enter the stage for processing. Records that do not meet all the preconditions are processed based on the error handling configured for the stage. Click **Add** to create additional preconditions.
On Record Error	Error record handling for the stage: • Discard - Discards the record. • Send to Error - Sends the record to the pipeline for error handling. • Stop Pipeline - Stops the pipeline. Not valid for cluster pipelines.
Required Fields	Field that must include data for the record to be passed into the stage. In a previous stage, you might want to first insert the fields that trigger the current stage. Records that do not include all the required fields are processed based on the error handling configured for the pipeline.
Description	Optional description.
Name	Stage name.

Configuration tab

WordNet Extractor Configuration Property	Description
Input Fields	Fields to process
Category Filter type	Choose what categories should be present in the output. Filters the categories based on the following options: • None - no filter • Blacklist - Categories specified here are excluded from the output. • Whitelist - Only categories specified here are included from the output.
Lowercase text	If checked, input text is converted to lowercase before the attribute extraction begins.
Use hierarchical categorical naming	If checked, all found attribute categories are prefixed or suffixed based on the **Hierarchical name location** by their field name using the separator specified in the **Hierarchical naming separator**. See Example of property "Use hierarchical category naming".
Hierarchical name location	Specifies where to add the hierarchical naming information. • Place before category name - Places the field name as a prefix to the attribute category with the **Hierarchical naming separator** as a separator between the field name and the attribute category. • Place after category name - Places the field name as a suffix to the attribute category with the **Hierarchical naming separator** as a separator between the field name and the attribute category.
Hierarchical naming separator	Separator between the field name and the attribute category.

Entity Selection tab

WordNet Extractor Entity Selection Property	Description
Short Categories	If checked, complete hypernym is truncated to base category. For example, verb.motion is truncated to verb.

Alphabetical Nouns only	If checked, words are classified as nouns only if they do not include numbers or special characters. Hyphens and underscores are allowed. If unchecked, nouns can be inferred from context, regardless of their letter content.
Word break characters	Characters to split strings to words.

Blacklisting and whitelisting categories

To configure the **Blacklist** tab or **Whitelist** property:

1. On the **Configuration** tab's **Category Filter Type**, select either **blacklist** or **whitelist**.
2. Configure the following desired properties:

WordNet Extractor Blacklist/Whitelist Property	Description
Blacklist	Specify categories that should be excluded from the output. For example, if `noun.artifact` is specified, this category is excluded from the output.
Whitelist	Specify categories here that should be included in the output. For example, if `noun.artifact` is specified, this category is included in the output.

EA StreamSets destinations--SMB resources and schemas

Note on EA deployment: The destinations for the StreamSets pipelines were resources defined in SMB running on the ISAA platform, which are essentially database tables. The data in these tables was the basis for field resource schemas and input to the THOPs developed for EA.

The EA deployment included the following types of data. These data were defined in resource schemas that specify field definitions for the data that was stored in SMB resources as a result of ETL ingestion via StreamSets processors. For background on that process, see Flow of ETL ingestion pipelines in StreamSets Data Collector

- Customer resource schema
- SAR resource schema
- Case resource schema
- Transaction field resource schema, which encompasses the following types of transactions;
  - ATM transaction schema
  - Wire transfer schema
  - ACH transaction schema
  - Large cash transaction schema
  - Person-to-Person transaction schema
- A subset of fields from the customer resource schema are shared in the other resource schemas for faster queries by avoiding double table lookups. See Denormalized data fields shared across schemas

Schema format

The format of the schemas is AVRO/JSON. However, for some of the schemas detailed below, rather than AVRO/JSON, the schemas are presented as simple tables for easier reading.

Recommendation on field names in AML Advisor user interface

Many of the field names defined in the schemas are visible in the AML Advisor user interface, as mentioned in the ISAA AML Advisor Guide, EA edition. For this reason, it is a best practice to define field names that are obviously and clearly understandable to your users.

Customer resource schema

The customer resource schema includes field definitions that identify specific customers.

Field Name	Nested Field Name	Example Value	Source Table	Notes

AML_PTY_ID		123456789	CPV_PTY_DEMOG_PROFL_CUR	Key value used across multple tables
PTY_ID		12345678912	CPV_PTY_DEMOG_PROFL_CUR	
GCI_Number		null (90%)	CPV_PTY_DEMOG_PROFL_CUR	
Coper_ID		null (90%)	CPV_PTY_DEMOG_PROFL_CUR	
SSN		123456789	CPV_PTY_DEMOG_PROFL_CUR	
Employer_ID_number		null (90%)	CPV_PTY_DEMOG_PROFL_CUR	
MLID		null (90%)	CPV_PTY_DEMOG_PROFL_CUR	
Primary_Party_ID_Number		123456789	CPV_PTY_DEMOG_PROFL_CUR	
Birth_date		1980-01-01	CPV_PTY_DEMOG_PROFL_CUR	
Country_of_Citizenship		US	CPV_PTY_DEMOG_PROFL_CUR	
Country_of_Residence		US	CPV_PTY_DEMOG_PROFL_CUR	
First_Name		Bob	CPV_PTY_DEMOG_PROFL_CUR	
Last_Name		Smith	CPV_PTY_DEMOG_PROFL_CUR	
Customer_TNUR_Date		2001-11-01	CPV_PTY_DEMOG_PROFL_CUR	
Email_Address		USA	CPV_PTY_DEMOG_PROFL_CUR	
NAICS_Occupation_Code		null (90%)	CPV_PTY_DEMOG_PROFL_CUR	
Primary_party_id_type		foreign passport	CPV_PTY_DEMOG_PROFL_CUR	
Primary_party_id_Issue_tx		ET	CPV_PTY_DEMOG_PROFL_CUR	
Primary_party_id_issue_date		2010-11-01	CPV_PTY_DEMOG_PROFL_CUR	
Primary_party_id_exp		2018-11-01	CPV_PTY_DEMOG_PROFL_CUR	
Primary_party_id_issue_State_province		AZ	CPV_PTY_DEMOG_PROFL_CUR	
primary_party_id_country		US	CPV_PTY_DEMOG_PROFL_CUR	
Organization_name		null	CPV_PTY_DEMOG_PROFL_CUR	
Ta_Line_of_Business		ABCDE	CPV_PTY_DEMOG_PROFL_CUR	
ppl_ofc_Country_Code		null	CPV_PTY_DEMOG_PROFL_CUR	
Occupation_desc		Labor	CPV_PTY_FIN_PROFL_CUR:	
Employer_name		mack's construction	CPV_PTY_FIN_PROFL_CUR:	
source_of_income		employment income	CPV_PTY_FIN_PROFL_CUR:	
net_worth		null (90%)	CPV_PTY_FIN_PROFL_CUR:	
Source_wealth_description		null (90% same value)	CPV_PTY_FIN_PROFL_CUR:	
Address				JSON List to show all addresses
	address_type	physical	CPV_PTY_ADDR_CUR:	
	address_line_1	123 happy st	CPV_PTY_ADDR_CUR:	
	address_line_2	null (90%)	CPV_PTY_ADDR_CUR:	
	city	Cary	CPV_PTY_ADDR_CUR:	
	state_code	NC	CPV_PTY_ADDR_CUR:	
	state_name	North Carolina	CPV_PTY_ADDR_CUR:	
	postal_code	12345	CPV_PTY_ADDR_CUR:	
Phone				JSON list to show all phone numbers
	Phone_type	CELL	CPV_PTY_PHN_CUR:	
	Phone_Number	1234567890	CPV_PTY_PHN_CUR:	

Related_parties				JSON list to show all related AML party IDs
	Related_AML_party_id	123456789	CPV PTY_NTWRK_PROFL_CUR:	
	Related_type_id	house	CPV PTY_NTWRK_PROFL_CUR:	
Accounts				JSON list to show all accounts
	AML_ACC_ID	1234567890	CPV_ACC_DEMOG_PROFL_CUR	
	ACC_open_date	2010-11-1	CPV_ACC_DEMOG_PROFL_CUR	
	ACC_close_date	null (90%)	CPV_ACC_DEMOG_PROFL_CUR	

SAR resource schema

This resource schema represents the standard contents of a Suspicious Activity Report.

The customer-related field definitions in the SAR resource, in the block beginning `"name": "customers",` come from the Denormalized data fields shared across schemas.

```
{
    "type": "record",
    "namespace": "com.intel.saffron",
    "name": "sar",
    "fields": [
        { "name": "sar_id", "type" : "long" },
        { "name": "src_sys_cd", "type" : ["null", "string"] },
        { "name": "cas_id", "type" : ["null", "string"] },
        { "name": "typ_cd", "type" : ["null", "string"] },
        { "name": "sar_stat_cd", "type" : ["null", "string"] },
        { "name": "sar_strt_dt", "type" : ["null", "string"] },
        { "name": "sar_end_dt", "type" : ["null", "string"] },
        { "name": "vlat_fr_dt", "type" : ["null", "string"] },
        { "name": "vlat_end_dt", "type" : ["null", "string"] },
        { "name": "sar_prep_dt", "type" : ["null", "string"] },
        { "name": "sar_rview_dt", "type" : ["null", "string"] },
        { "name": "aprv_dt", "type" : ["null", "string"] },
        { "name": "sar_fi_dt", "type" : ["null", "string"] },
        { "name": "sar_dcn", "type" : ["null", "string"] },
        { "name": "sar_fi_by_id", "type" : ["null", "string"]},
        { "name": "rlt_dcn", "type" : ["null", "string"] },
        { "name": "aprv_by_wrk_id", "type" : ["null", "string"] },
        { "name": "bsa_trk_id", "type" : ["null", "string"] },
        { "name": "actvy_am", "type" : ["null", "string"] },
        { "name": "actvy_loss_am", "type" : ["null", "string"] },
        { "name": "matl_impac_in", "type" : ["null", "string"] },
        { "name": "cntct_nm", "type" : ["null", "string"] },
        { "name": "cnct_phn_no", "type" : ["null", "string"] },
        { "name": "cntct_phn_ext", "type" : ["null", "string"] },
        { "name": "cntct_titl_nm", "type" : ["null", "string"] },
        { "name": "rfi_ent_nm", "type" : ["null", "string"] },
        { "name": "rfi_ein", "type" : ["null", "string"] },
        { "name": "rfi_addr_tx", "type": ["null", "string"] },
        { "name": "rfi_cty_nm", "type": ["null", "string"] },
```

```
        { "name": "rfi_st_cd", "type": ["null", "string"] },
        { "name": "rfi_zip_cd", "type": ["null", "long"] },
        { "name": "xmit_nm", "type": ["null", "string"] },
        { "name": "xmit_addr_tx", "type": ["null", "string"] },
        { "name": "xmit_cty_nm", "type": ["null", "string"] },
        { "name": "xmit_st_cd", "type": ["null", "string"] },
        { "name": "xmit_zip_cd", "type": ["null", "string"] },
        { "name": "xmit_phn_no", "type": ["null", "string"] },
        { "name": "cntct_agncy_nm", "type": ["null", "string"] },
        { "name": "agncy_cntct_phn_ext", "type": ["null", "string"] },
        { "name": "src_last_updated_by", "type": ["null", "string"] },
        { "name": "pop_batch_id", "type": ["null", "int"] },
        { "name": "exp_batch_id", "type": ["null", "int"] },
        { "name": "cas_dt", "type": ["null", "string"] },
        { "name": "dsptn_id", "type": ["null", "int"] },
        { "name": "customers", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "customer", "type": "record",
                    "fields": [
                        {"name": "aml_pty_id",  "type" : "long" },
                        { "name": "pty_id", "type" : ["null", "string"] },
                        { "name": "gci_no", "type" : ["null", "string"] },
                        { "name": "coper_id", "type" : ["null", "string"]
},
                        { "name": "ssn", "type" : ["null", "string"] },
                        { "name": "ein", "type" : ["null", "string"] },
                        { "name": "mlid", "type" : ["null", "string"] },
                        { "name": "brth_dt", "type" : ["null", "string"] },
                        { "name": "cnty_of_ctzn_cd", "type" : ["null",
"string"] },
                        { "name": "cnty_of_rsdn_cd", "type" : ["null",
"string"] },
                        { "name": "frst_nm", "type" : ["null", "string"] },
                        { "name": "lst_nm", "type" : ["null", "string"] },
                        { "name": "eml_addr_tx", "type" : ["null",
"string"] },
                        { "name": "naics_cd", "type" : ["null", "string"]
},
                        { "name": "ta_lob_in", "type" : ["null", "string"]
},
                        { "name": "ocptn_de", "type": ["null", "string"] },
                        { "name": "empr_nm", "type": ["null", "string"] },
                        { "name": "src_of_incm", "type": ["null", "string"]
},
                        { "name": "net_wrth", "type": ["null", "long"] },
                        { "name": "phones", "type": ["null", {
                            "type": "array",
                                "items" : {
                                    "name": "phone", "type": "record",
                                    "fields": [
                                        {"name": "phn_typ", "type":
["null", "string"]},
```

```
                                        {"name": "phn_no", "type":
["null", "string"]}
                                    ]
                                }
                            }]
                        },
                        { "name": "addresses", "type": ["null", {
                            "type":  "array",
                                "items" : {
                                    "name": "address", "type": "record",
                                    "fields": [
                                        {"name": "addr_typ", "type":
["null", "string"]},
                                        {"name": "addr_lin_1_tx", "type":
["null", "string"]},
                                        {"name": "addr_lin_2_tx", "type":
["null", "string"] },
                                        {"name": "cty_nm", "type":
["null", "string"]},
                                        {"name": "cntry_cd", "type":
["null", "string"]},
                                        {"name": "pst_cd", "type":
["null", "string"]},
                                        {"name": "st_of_prvnc_cd", "type":
["null", "string"]},
                                        {"name": "st_or_prvnc_nm", "type":
["null", "string"]}
                                    ]
                                }
                            }]
                        },
                        { "name": "accounts", "type": ["null", {
                            "type": "array",
                                "items" : {
                                    "name": "account", "type": "record",
                                    "fields": [
                                        {"name": "aml_acc_id", "type":
"long"},
                                        {"name": "acc_opn_dt", "type":
["null", "string"]},
                                        {"name": "acc_cls_dt", "type":
["null", "string"] }
                                    ]
                                }
                            }]
                        }]
                }
            }]
        },
        { "name": "narratives", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "narrative", "type": "record",
```

```
                    "fields": [
                        {"name": "nrrtv_tx", "type": ["null", "string"]},
                        {"name": "typ_cd", "type": ["null", "string"]}
                    ]
                }
            }]
        },
        { "name": "branches", "type": ["null", {
            "type":  "array",
                "items" : {
                    "name": "branch", "type": "record",
                    "fields": [
                        {"name": "sar_brn_seq_no", "type": ["null",
"int"]},
                        {"name": "typ_cd", "type": ["null", "string"] },
                        {"name": "brn_cd", "type": ["null", "string"]},
                        {"name": "brn_addr_tx", "type": ["null",
"string"]},
                        {"name": "brn_cty_nm", "type": ["null", "string"]},
                        {"name": "brn_st_cd", "type": ["null", "string"]},
                        {"name": "brn_zip_cd", "type": ["null", "string"]},
                        {"name": "brn_cntry_cd", "type": ["null",
"string"]},
                        {"name": "sspcs_actvy_brn_in", "type": ["null",
"string"]},
                        {"name": "moc_brn_id", "type": "long"},
                        {"name": "brn_id", "type": ["null", "long"]},
                        {"name": "brn_cntry", "type": ["null", "string"]},
                        {"name": "brn_cc_no", "type": ["null", "string"]},
                        {"name": "brn_co_no", "type": ["null", "string"]},
                        {"name": "brn_addr1", "type": ["null", "string"]},
                        {"name": "brn_addr2", "type": ["null", "string"]},
                        {"name": "brn_cty", "type": ["null", "string"]},
                        {"name": "brn_st", "type": ["null", "string"]},
                        {"name": "brn_zip", "type": ["null", "string"]},
                        {"name": "mn_phn_num", "type": ["null", "string"]},
                        {"name": "sec_phn_num", "type": ["null",
"string"]},
                        {"name": "mgr_phn_num", "type": ["null",
"string"]},
                        {"name": "asst_mgr_phn_num", "type": ["null",
"string"]},
                        {"name": "atm_locat_othr_desc", "type": ["null",
"string"]},
                        {"name": "atm_locat_cd", "type": ["null",
"string"]},
                        {"name": "atm_cam_ind", "type": ["null",
"string"]},
                        {"name": "entr_typ_ind", "type": ["null",
"string"]}
                    ]
                }
            }]
```

```
        },
        { "name": "parties", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "party", "type": "record",
                    "fields": [
                        {"name": "pty_id", "type":  ["null", "string"] },
                        {"name": "aml_pty_id", "type": "long" },
                        {"name": "acc_pd_ctgy_cd", "type": ["null",
"string"] },
                        {"name": "gci_no", "type": ["null", "string"] },
                        {"name": "acc_appsys_id", "type": ["null",
"string"] },
                        {"name": "acc_co_no", "type": ["null", "string"] }
                    ]
                }
            }]
        },
        { "name": "events", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "event", "type": "record",
                    "fields": [
                        {"name": "case_id", "type": ["null", "string"]},
                        {"name": "ev_id", "type": "long"},
                        {"name": "is_suspect", "type": ["null", "string"]
},
                        {"name": "moc_ev_id", "type": ["null", "int"] }
                    ]
                }
            }]
        },
        { "name": "journals", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "journal", "type": "record",
                    "fields": [
                        {"name": "jrnl_tx", "type": ["null", "string"]},
                        {"name": "typ_cd", "type": ["null", "string"]}
                    ]
                }
            }]
        },
        { "name": "frm_non_rpts", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "frm_non_rpt", "type": "record",
                    "fields": [
                        {"name": "cas_id", "type": ["null", "string"]},
                        {"name": "dsptn_id", "type": ["null", "long"]},
                        {"name": "rjct_cmt", "type": ["null", "string"]},
                        {"name": "stat", "type": ["null", "string"]},
                        {"name": "reg_nm", "type": ["null", "string"]}
```

```
                ]
            }
        }]
```

```
        }
    ]
}
```

Case resource schema

This schema represents the contents of a typical AML investigation case.

The customer-related field definitions in the case resource, in the block beginning `"name": "customers"`, come from the Denormalized data fields shared across schemas.

```
{
    "type": "record",
    "namespace": "com.intel.saffron",
    "name": "case_investigation",
    "fields": [
        { "name": "moc_cas_id", "type" : "long" },
        { "name": "cas_id", "type" : ["null", "string"] },
        { "name": "rsn_cd", "type" : ["null", "string"] },
        { "name": "opn_dt", "type" : ["null", "string"] },
        { "name": "cls_dt", "type" : ["null", "string"] },
        { "name": "cas_prty_cd", "type" : ["null", "string"] },
        { "name": "cas_stat_cd", "type" : ["null", "string"] },
        { "name": "cas_efect_dt", "type" : ["null", "string"] },
        { "name": "ep_ev_grp_id", "type" : ["null", "string"] },
        { "name": "prmy_lob", "type" : ["null", "string"] },
        { "name": "investigator_first_name", "type" : ["null", "string"] },
        { "name": "investigator_middle_name", "type" : ["null", "string"]
},
        { "name": "investigator_last_name", "type" : ["null", "string"] },
        { "name": "stry_typ_cd", "type" : ["null", "string"] },
        { "name": "stry_txt", "type" : ["null", "string"] },
        { "name": "risk_factors", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "risk_factor", "type": "record",
                    "fields": [
                        { "name": "rsk_fctr_id", "type" : ["null",
"string"] },
                        { "name": "subj_typ_cd", "type" : ["null",
"string"] },
                        { "name": "subj_id", "type" : ["null", "string"] },
                        { "name": "rsk_fctr_cd", "type" : ["null",
"string"] },
                        { "name": "rsk_fctr_de", "type" : ["null",
"string"] },
                        { "name": "rsk_score", "type" : ["null", "string"]
},
                        { "name": "rsk_action", "type" : ["null", "string"]
}
```

```
                    ]
                }
            }]
        },

        { "name": "narratives", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "narrative", "type": "record",
                    "fields": [
                        {"name": "typ_cd", "type": ["null", "string"]},
                        {"name": "nrrtv_tx", "type": ["null", "string"]}
                    ]
                }
            }]
        },
        { "name": "case_accounts", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "case_account", "type": "record",
                    "fields": [
                        {"name": "arng_id", "type": "long"},
                        {"name": "accno", "type": ["null", "string"]},
                        {"name": "acc_pd_ctgy_cd", "type": ["null",
"string"]},
                        {"name": "acc_appsys_id", "type": ["null",
"string"]},
                        {"name": "acc_co_no", "type": "int"}
                    ]
                }
            }]
        },
        { "name": "case_parties", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "case_party", "type": "record",
                    "fields": [
                        {"name": "pty_id", "type": ["null", "string"]},
                        {"name": "pty_typ_cd", "type": ["null", "string"]},
                        {"name": "pty_tax_id", "type": ["null", "string"]},
                        {"name": "pty_empr_nm", "type": ["null", "string"]
},
                        {"name": "pty_ocptn_nm", "type": ["null", "string"]
},
                        {"name": "pty_bus_indst_typ_cd", "type": ["null",
"string"] },
                        {"name": "pty_dgr_of_sep_in", "type": ["null",
"string"]},
                        {"name": "pty_lob_cd", "type": ["null", "string"]},
                        {"name": "pty_wlth_src_de", "type": ["null",
"string"] },
                        {"name": "pty_incm_src_de", "type": ["null",
"string"]},
```

```
                        {"name": "pty_sspt_in", "type": ["null",
"string"]},
                        {"name": "pty_gndr_cd", "type": ["null", "string"]
},
                        {"name": "pty_cls_rsn_cd", "type": ["null",
"string"] },
                        {"name": "frgn_pty_id", "type": ["null",
"string"]},
                        {"name": "aml_pty_id", "type": "long"}
                    ]
                }
            }]
        },
        { "name": "related_cases", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "related_case", "type": "record",
                    "fields": [
                        {"name": "rlt_cas_id", "type": ["null", "string"]},
                        {"name": "rel_typ_cd", "type": ["null", "string"]}
                    ]
                }
            }]
        },
        { "name": "case_phones", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "case_phone", "type": "record",
                    "fields": [
                        {"name": "moc_pty_id", "type": "int"},
                        {"name": "pty_id", "type": ["null", "string"]},
                        {"name": "phn_seq_no", "type": "int"},
                        {"name": "pty_phn_no", "type": ["null", "string"]},
                        {"name": "pty_phn_extsn_no", "type": ["null",
"string"] },
                        {"name": "aml_pty_id", "type": "long"}
                    ]
                }
            }]
        },
        { "name": "case_addresses", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "case_address", "type": "record",
                    "fields": [
                        {"name": "addr_seq_no", "type": "int"},
                        {"name": "pty_id", "type": ["null", "string"]},
                        {"name": "pty_addr_lin_1_tx", "type": ["null",
"string"]},
                        {"name": "pty_addr_lin_2_tx", "type": ["null",
"string"] },
                        {"name": "pty_cty_nm", "type": ["null", "string"]},
                        {"name": "pty_st_cd", "type": ["null", "string"]},
```

```
                        {"name": "pty_zip_cd", "type": ["null", "string"]},
                        {"name": "pty_cntry_cd", "type": ["null",
"string"]},
                        {"name": "aml_pty_id", "type": "long"}
                    ]
                }
            }]
        },
        { "name": "case_identifications", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "case_identification", "type": "record",
                    "fields": [
                        {"name": "pty_id", "type": ["null", "string"]},
                        {"name": "pty_id_mthd_of_vrfn_nm", "type": ["null",
"string"] },
                        {"name": "pty_id_typ_desc", "type": ["null",
"string"]},
                        {"name": "id_seq_no", "type": "int"},
                        {"name": "isu_st_cd", "type": ["null", "string"]},
                        {"name": "pty_id_isu_auth_cd", "type": ["null",
"string"]},
                        {"name": "isu_cntry_cd", "type": ["null",
"string"]},
                        {"name": "isu_dt", "type": ["null", "string"]},
                        {"name": "pty_id_exp_dt", "type": ["null",
"string"]},
                        {"name": "aml_pty_id", "type": "long"}
                    ]
                }
            }]
        },
        { "name": "case_aliases", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "case_alias", "type": "record",
                    "fields": [
                        {"name": "pty_id", "type": ["null", "string"]},
                        {"name": "alias_seq_no", "type": "int"},
                        {"name": "pty_alias_nm", "type": ["null", "string"]
},
                        {"name": "aml_pty_id", "type": "long"}
                    ]
                }
            }]
        },
        { "name": "case_emails", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "case_email", "type": "record",
                    "fields": [
                        {"name": "pty_id", "type": ["null", "string"]},
                        {"name": "eml_seq_no", "type": "int"},
```

```
                        {"name": "pty_eml_addr_tx", "type": ["null",
"string"]},
                        {"name": "aml_pty_id", "type": "long"}
                    ]
                }
            }]
        },
        { "name": "case_urls", "type": ["null", {
            "type": "array",
                "items" : {
                    "name": "case_url", "type": "record",
                    "fields": [
                        {"name": "pty_id", "type": ["null", "string"]},
                        {"name": "url_seq_no", "type": "int"},
                        {"name": "pty_url_tx", "type": ["null", "string"]
},
                        {"name": "aml_pty_id", "type": "long"}
                    ]
                }
            }]
        },
    { "name": "customers", "type": ["null", {
                "type": "array",
                    "items" : {
                        "name": "customer", "type": "record",
                        "fields": [
                            {"name": "aml_pty_id",  "type" : "long" },
                            { "name": "pty_id", "type" : ["null", "string"]
},
                            { "name": "gci_no", "type" : ["null", "string"]
},
                            { "name": "coper_id", "type" : ["null",
"string"] },
                            { "name": "ssn", "type" : ["null", "string"] },
                            { "name": "ein", "type" : ["null", "string"] },
                            { "name": "mlid", "type" : ["null", "string"]
},
                            { "name": "brth_dt", "type" : ["null",
"string"] },
                            { "name": "cnty_of_ctzn_cd", "type" : ["null",
"string"] },
                            { "name": "cnty_of_rsdn_cd", "type" : ["null",
"string"] },
                            { "name": "frst_nm", "type" : ["null",
"string"] },
                            { "name": "lst_nm", "type" : ["null", "string"]
},
                            { "name": "eml_addr_tx", "type" : ["null",
"string"] },
                            { "name": "naics_cd", "type" : ["null",
"string"] },
                            { "name": "ta_lob_in", "type" : ["null",
"string"] },
```

```
                            { "name": "ocptn_de", "type": ["null",
"string"] },
                            { "name": "empr_nm", "type": ["null", "string"]
},
                            { "name": "src_of_incm", "type": ["null",
"string"] },
                            { "name": "net_wrth", "type": ["null", "long"]
},
                            { "name": "phones", "type": ["null", {
                                "type": "array",
                                    "items" : {
                                        "name": "phone", "type": "record",
                                        "fields": [
                                            {"name": "phn_typ", "type":
["null", "string"]},
                                            {"name": "phn_no", "type":
["null", "string"]}
                                        ]
                                    }
                                }]
                            },
                            { "name": "addresses", "type": ["null", {
                                "type":  "array",
                                    "items" : {
                                        "name": "address", "type":
"record",
                                        "fields": [
                                            {"name": "addr_typ", "type":
["null", "string"]},
                                            {"name": "addr_lin_1_tx",
"type": ["null", "string"]},
                                            {"name": "addr_lin_2_tx",
"type": ["null", "string"] },
                                            {"name": "cty_nm", "type":
["null", "string"]},
                                            {"name": "cntry_cd", "type":
["null", "string"]},
                                            {"name": "pst_cd", "type":
["null", "string"]},
                                            {"name": "st_of_prvnc_cd",
"type": ["null", "string"]},
                                            {"name": "st_or_prvnc_nm",
"type": ["null", "string"]}
                                        ]
                                    }
                                }]
                            },
                            { "name": "accounts", "type": ["null", {
                                "type": "array",
                                    "items" : {
                                        "name": "account", "type":
"record",
                                        "fields": [
```

```
                                        {"name": "aml_acc_id", "type":
"long"},
                                        {"name": "acc_opn_dt", "type":
["null", "string"]},
                                        {"name": "acc_cls_dt", "type":
["null", "string"] }
                                    ]
                                }
                            }]
                        }]
                    }
                }]
```

```
            }
        ]
}
```

Transaction resource schema

This schema represents a generalized banking transaction that encompasses several different types of transactions:

- ATM transaction schema
- Wire transfer schema
- ACH transaction schema
- Large cash transaction schema
- Person-to-Person transaction schema

Transaction common attributes

These attributes are common to all transaction types.

Saffron Field Name	Explanation	Notes / Example values
isaa_type	ISAA Resource Type (generated by data ingestion)	"Transaction"
isaa_id	Unique ID used by platform (generated by data ingestion)	"transaction-<txn_type>-<txn_id>-<pst_txn_id>" For example - "transaction-wire-542854738-4287432"
txn_type	Type of transaction	"wire", "ach", "atm", "billpay"
txn_id	ID	"tran_id" in most of customer's tables
pst_txn_id	Post transaction ID	"pst_tran_id" in most of customer's tables, can be null
txn_am	Value amount of transaction in US dollars This to be the primary value the system operates on.	"tran_am" in some types, "tran_usd_eqvln_am" in others This should be a JSON float with 2 decimal places. **<QUESTION to Gary: Specify Scaled Float on Platform Ingest. WHERE?>**
curr_cd	Currency transaction was executed in	THree-character currency code defined by ISO 4217. See https://www.iso.org/iso-4217-currency-codes.html
curr_am	Amount in the currency of the transaction	If the curr_cd is USD, then curr_am is the same usd_am. This should be a JSON float with 2 decimal places. **<QUESTION to Gary: Specify Scaled Float on Platform Ingest. WHERE?>**
txn_dt	Date of transaction in "yyyy-MM-dd" format	
pst_txn_dt	Date the transaction posted.	
txn_sub_type	Transaction sub type	Optional. Usally maps to a category field.
txn_extra_info	Transaction extra info	Optional.
txn_tm	Time of Transaction, in HHmmss format	

cr_dr_in	Credit/debit indicator	'C' for credit, 'D' for debit. A Credit/debit indicator, added for aggregation purposes, because the customer's data has duplicated rows for certain transaction types. Not all transaction types use this - depending on how the data is represented.

Transaction schema credit and debit party attributes

An ISAA transaction resource has one or two parties: a credit party and a debit party. A party can be null, depending on the type of transaction; for example, an ATM deposit has no debit party.

The fields for both parties are the same for credit or debit. This is noted in the schema definition with **cr**_ or **dr**_. See the field cr_dr_in in Wire transfer schema below.

> The schema distinguishes between a *customer account* location and a *transaction's* location. For example, a customer might take his US California ATM card (the account's location) on a trip and withdraw money from an ATM in Yemen (the transaction's location).

Saffron Field Name	Explanation	Notes / Example values
cr_acc_no	Credit Account Number	"tran_accno" or "cr_accno", depending on transaction type and attributes
cr_acc_co_no	Credit Account Company Number	"tran_acc_co_no" or "cr_acc_co_no"
cr_acc_prod_ctgy_cd	Credit Account Product Category Code	
cr_acc_addr	Credit Account Address	If available - merge multiple lines, or have mulitple addr_1, addr_2 lines?
cr_acc_cty	Credit Account city	If available
cr_acc_st_cd	Credit Account State Code	If available - for Out of State txn agggregations
cr_acc_pst_cd	Credit Account postal code/zip code	If available
cr_acc_cntry_cd	Credit Account Country code	If available
cr_acc_geocode	QUESTION Stretch goal feature	
cr_txn_addr	Credit Transaction Address	If available - merge multiple lines, or have mulitple addr_1, addr_2 lines?
cr_txn_cty	Credit Transaction City	If available
cr_txn_st_cd	Credit Transaction State Code	If available - for Out of State txn agggregations
cr_txn_pst_cd	Credit Transaction postal code/zip code	If available
cr_txn_cntry_cd	Credit Transaction Country code	If available
cr_txn_geocode	QUESTION Stretch goal feature	
cr_customer	From Denormalized data fields shared across schemas	Nested, joined via cr_acc_no -> aml_acc_id -> aml_pty_id, usually.

ATM transaction schema

Saffron Field Name	Explanation	Notes / Example values
isaa_type	ISAA Resource Type (generated by data ingestion)	"Transaction"
isaa_id	Unique ID used by platform (generated by data ingestion)	"transaction-atm-<chn_tran_id>"
txn_type	Type of transaction	"atm"

txn_id	ID	"chn_tran_id"
pst_txn_id	Post transaction ID	"pst_tran_id"
txn_am	Value amount of transaction in US dollars This is the primary field the system operaties on.	"tran_curr_am"
curr_cd	Currency transaction was executed in	"tran_locl_curr_cd"
curr_am	Amount in the currency of the transaction	"tran_locl_curr_am"
txn_dt	Date of transaction in "yyyy-MM-dd" format	"chn_src_tran_dt"
pst_txn_dt	Date the transaction posted.	"tran_pst_dt"
txn_sub_type	Transaction sub type	chn_tran_typ_ctgy_cd - "Deposit, withdrawal, etc"
txn_extra_info	Transaction extra info	"chn_tran_typ_de"
txn_tm	Time of transaction, in HHmmss format	"chn_src_tran_tm"

Saffron Field Name	Explanation	Notes / Example values
cr_acc_no	Credit Account Number	"tran_accno" / "cdt_accno" / "dbt_accno" depending explanation above
cr_acc_co_no	Credit account Company Number (for companies, not persons, as far as I can tell)	"tran_acc_co_no" / "cdt_acc_co_no" / "dbt_acc_co_no"
cr_acc_prod_ctgy_cd	Credit account Product Category Code	"tran_acc_prod_ctgy_cd" (can't find a cdt / dbt version)
cr_acc_addr	Credit account Address	null
cr_acc_cty	Credit account city	null
cr_acc_st_cd	Credit account State Code	"chn_crd_st_cd" - use for out of state ATM calculation. Need to validate
cr_acc_pst_cd	Credit account postal code/zip code	null
cr_acc_cntry_cd	Credit account Country code	null
cr_acc_geocode	QUESTION: Stretch goal feature	null
cr_txn_addr	Credit Transaction Address	"atm_trmnl_addr_lin_1_tx" "atm_trmnl_addr_lin_2_tx"
cr_txn_cty	Credit Transaction City	atm_cty_nm
cr_txn_st_cd	Credit Transaction State Code	"atm_st_cd" - this always has states. atm_trmnl_st_cd has some weird fields sometimes)
cr_txn_pst_cd	Credit Transaction postal code/zip code	atm_zip_cd
cr_txn_cntry_cd	Credit Transaction Country code	atm_cntry_cd
cr_txn_geocode	QUESTION Stretch goal feature	
cr_customer	From Denormalized data fields shared across schemas	Join from tran_accno / cdt_accno / dbt_accno, depending on above explanation

ATM transaction metadata

The "txn_metadata" field includes the following:

chn_crd_typ_cd
chn_crd_isu_cd
chn_crd_no

chn_crd_brnd_nm
chn_bin_de
atm_trmnl_id
atm_trmnl_fi_nm
atm_trmnl_ntwk_id
atm_trmnl_ntwrk_nm
atm_rvrsl_rsn_cd
atm_rvrsl_rsn_de_tx

Wire transfer schema

Saffron Field Name	Explanation	Notes / Example values
isaa_type	ISAA resource type (generated by data ingestion)	"Transaction"
isaa_id	Unique ID used by platform (generated by data ingestion)	"transaction-atm-<tran_id>-<pst_tran_id>"
txn_type	Type of transaction	"wire"
txn_id	ID	"tran_id"
pst_txn_id	Post transaction ID	"pst_tran_id", <-- 10/27 update - this has to be pst_tran_id. ~~In the table, "src_tran_id" looks to be always populated.~~ ~~**Do we need to add that field here?**~~
txn_am	Value amount of transaction in US dollars This is the primary field the system operates on.	"tran_usd_eqvln_am"
curr_cd	Currency transaction was executed in	"tran_curr_cd"
curr_am	Amount in the currency of the transaction	"tran_curr_am"
txn_dt	Date of transaction in "yyyy-MM-dd" format	"orig_tran_dt"
pst_txn_dt	Date the transaction posted.	"src_tran_dt" is always populated. Do we need to add that?
txn_sub_type	Transaction sub type	"tran_typ_cd" - FTR, DEBIT, PRE, CREDIT, SPL are some of the values here
txn_extra_info	Transaction extra info	"src_sys_id" - WTX, GBS-ARIES, or FPS
txn_tm	Time of transaction, in HHmmss format	"wir_init_tm" - wire initiation time
cr_dr_in	Credit/debit indicator	QUESTION: Still to be finalized, pending further understanding of how the bank represents their data. If there are two rows for one wire transaction, this can be used in aggregations, without double counting.

Wire transfer transaction credit and debit party attributes

Both debit and credit parties always exist.

Saffron Field Name	Explanation	Notes / Example values
cr_acc_no	Credit account number	"dr_accno" and "cr_accno" - always 2 parties

cr_acc_co_no	Credit account company number (for companies, not persons, as far as I can tell)	"dr_acc_co_no" and "cr_acc_co_no"
cr_acc_prod_ctgy_cd	Credit account product category code	"dr_acc_prd_ctgy_cd" and "cr_acc_prod_ctgy_Cd"
cr_acc_addr	Credit account address	combination of "cr_pty_addr_1_tx", 2_tx, and 3_tx
cr_acc_cty	Credit account city	null
cr_acc_st_cd	Credit account state code	null
cr_acc_pst_cd	Credit account postal code/zip code	null
cr_acc_cntry_cd	Credit account country code	"dr_pty_cntry_cd" and "cr_pty_cntry_cd"
cr_acc_geocode	Stretch goal feature	
cr_txn_addr	Credit transaction address	None - we don't know what a transaction locatoin is yet
cr_txn_cty	Credit transaction city	None - we don't know what a transaction locatoin is yet
cr_txn_st_cd	Credit transaction state code	None - we don't know what a transaction locatoin is yet
cr_txn_pst_cd	Credit transaction postal code/zip code	None - we don't know what a transaction locatoin is yet
cr_txn_cntry_cd	Credit transaction country code	None - we don't know what a transaction locatoin is yet
cr_txn_geocode	Stretch goal feature	
cr_customer	From Denormalized data fields shared across schemas	Nested and usually joined via cr_acc_no -> aml_acc_id -> aml_pty_id.

Wire transfer transaction metadata

The "txn_metadata" field includes the following:

dr_acc_nm
cr_acc_nm
instm_tran_acc_brn_addr
instm_tran_acc_brn_nm
instm_tran_acc_brn_no
instm_tran_acc_cc_no
bnf_bnk_pty_id
bnf_bnk_pty_nm
bnf_bnk_pty_addr_1_tx
bnf_bnk_pty_addr_2_tx
bnf_bnk_pty_addr_3_tx
bnf_bnk_pty_addr_4_tx
bnf_bnk_pty_cntry_cd
bnf_pty_id
bnf_pty_nm
bnf_pty_addr_1_tx
bnf_pty_addr_2_tx
bnf_pty_addr_3_tx

bnf_pty_cntry_cd
cr_pty_nm
cr_pty_addr_1_tx
cr_pty_addr_2_tx
cr_pty_addr_3_tx
cr_pty_cntry_cd
dr_pty_nm
dr_pty_addr_1_tx
dr_pty_addr_2_tx
dr_pty_addr_3_tx
dr_pty_cntry_cd
ibk_pty_id
ibk_pty_nm
ibk_bnk_pty_addr_1_tx
ibk_bnk_pty_addr_2_tx
ibk_bnk_pty_addr_3_tx
ibk_bnk_pty_addr_4_tx
ibk_pty_cntry_cd
ord_bnk_pty_id
ord_bnk_pty_nm
ord_bnk_pty_addr_1_tx
ord_bnk_pty_addr_2_tx
ord_bnk_pty_addr_3_tx
ord_bnk_pty_addr_4_tx
ord_bnk_pty_cntry_cd
orig_pty_id
orig_pty_nm
orig_pty_addr_1_tx
orig_pty_addr_2_tx
orig_pty_addr_3_tx
orig_pty_cntry_cd
orig_pty_to_bnf_instr_1_tx
orig_pty_to_bnf_instr_2_tx
orig_pty_to_bnf_instr_3_tx
orig_pty_to_bnf_instr_4_tx
snd_bnk_pty_acc_no
snd_bnk_pty_id

snd_bnk_pty_nm
snd_bnk_pty_addr_1_tx
snd_bnk_pty_addr_2_tx
snd_bnk_pty_addr_3_tx
snd_bnk_pty_cntry_cd

ACH transaction schema

The ACH (Automated Clearing House) transaction data are in three tables:

- btx_tran_vw_ach_pepplus
- btx_tran_acc_vw_ach_pepplus
- btx_sub_tran_instm_vw_ach_pepplus

These tables are joined by tran_id and by sub_tran_seq_no.

ACH common transaction attributes

Saffron Field Name	Explanation	Notes / Example values
isaa_type	ISAA resource type (generated via data ingestion)	"Transaction"
isaa_id	Unique ID used by platform (generated via data ingestion)	"transaction-<txn_type>-<txn_id>-<sub_tran_seq_no>" For example - "transaction-ach-542854738-2"
txn_type	Type of transaction	"ach"
txn_id	ID	"tran_id" . This is the key for joining in all 3 tables
pst_txn_id	Post transaction ID	"src_tran_id"
txn_am	Value amount of transaction in US dollars This is the primary field the system operates on.	"btx_sub_tran_instm_vw_ach_pepplus.instm_curr_am"
curr_cd	Currency transaction was executed in	"btx_sub_tran_instm_vw_ach_pepplus.instm_curr_cd"
curr_am	Amount in the currency of the transaction	null - none of the other amount fields are populated
txn_dt	Date of transaction in "yyyy-MM-dd" format	"btx_tran_vw_ach_pepplus.src_tran_dt"
pst_txn_dt	Date the transaction posted.	"btx_tran_vw_ach_pepplus.tran_pst_dt"
txn_sub_type	Transaction sub type	btx_tran_vw_ach_pepplus.tctgy_cd Some possible values "WEB", "TEL", "PPD", "CCD"
txn_extra_info	Transaction extra info	This should be the su- transaciton sequence number that is joined on "btx_sub_tran_instm_vw_ach_pepplus.sub_tran_seq_no"
txn_tm	Time of transaction, in HHmmss format	"src_tran_tm"

ACH credit and debit party attributes

If btx_sub_tran_instm_vw_ach_pepplus.sub_tran_typ_cd = 'DEBIT', then each of the sub-parties are 'dr', and the main party is 'cr'.

Conversely, if btx_sub_tran_instm_vw_ach_pepplus.sub_tran_typ_cd = 'CREDIT', then the sub-party is the 'cr', and the main party is 'dr'.

The location of the fields is the same for both: the account related information is in the btx_tran_acc_vw_ach_pepplus table.

Saffron Field Name	Explanation	Notes / Example values
(c|d)r_acc_no	Credit account number	btx_tran_acc_vw_ach_pepplus.tran_accno
(c|d)r_acc_co_no	Credit account company number (for companies, not persons, as far as I can tell)	btx_tran_acc_vw_ach_pepplus.tran_acc_co_no
(c|d)r_acc_prod_ctgy_cd	Credit account product category code	btx_tran_acc_vw_ach_pepplus.tran_acc_prod_ctgy_cd
(c|d)r_acc_addr	Credit account address	btx_tran_acc_vw_ach_pepplus.tran_acc_cc_no & tran_acc_co_no Look up using Location table for bank location.
(c|d)r_acc_cty	Credit account city	Part of above
(c|d)r_acc_st_cd	Credit account state code	Part of above
(c|d)r_acc_pst_cd	Credit account postal code/zip code	Part of above
(c|d)r_acc_cntry_cd	Credit account country code	Part of above
(c|d)r_acc_geocode	Stretch goal feature	Not available. Had to use the address of the account from Denormalized data fields shared across schemas.
(c|d)r_txn_addr	Credit transaction address	NULL
(c|d)r_txn_cty	Credit transaction city	NULL
(c|d)r_txn_st_cd	Credit transaction state code	NULL
(c|d)r_txn_pst_cd	Credit transaction postal code/zip code	NULL
(c|d)r_txn_cntry_cd	Credit transaction country code	NULL
(c|d)r_txn_geocode	Stretch goal feature	NULL
(c|d)r_customer	From	Nested, usually joined via cr_acc_no -> aml_acc_id -> aml_pty_id.

ACH transaction metadata

The "txn_metadata" field includes the following:

btx_tran_vw_ach_pepplus.src_sys_id
btx_tran_vw_ach_pepplus.orig_appsys_id
btx_tran_vw_ach_pepplus.tran_curr_cd
btx_tran_vw_ach_pepplus.tran_curr_am This is the total amount that the "main" party paid.
btx_tran_vw_ach_pepplus.tran_usd_eqvln_am (mostly null)
btx_tran_vw_ach_pepplus.tran_de_tx
btx_tran_acc_vw_ach_pepplus.tran_acc_bofa_in

Large cash transactions schema

Large cash common transaction attributes

Saffron Field Name	Explanation	Notes / Example values
isaa_type	ISAA Resource Type (generated by data ingestion)	"Transaction"
isaa_id	Unique ID used by platform (generated by data ingestion)	"transaction-<txn_type>-<txn_id>" For example - "transaction-lg_csh-542854738"
txn_type	Type of transaction	"lg_csh"
txn_id	ID	"tran_seq_no" // no tran_id - this is unique also
pst_txn_id	Post transaction ID	Null
txn_am	Value amount of transaction in US dollars This is the primary field the system operates on.	we need to consider both csh_in_am and csh_out_am So either: 1) ABS (csh_in_am - csh_out_am), or 2) if (csh_in_am > csh_out_am), csh_in_am - csh_out_am, else csh_out_am - csh_in_am Which is the same operation, really.
curr_cd	Currency transaction was executed in	No Currency indicator in data - assume all USD - there are deposits at the teller 'USD'
curr_am	Amount in the currency of the transaction	same as txn_am
txn_dt	Date of transaction in "yyyy-MM-dd" format	"tran_dt"
pst_txn_dt	Date the transaction posted	"pst_dt"
txn_sub_type	Transaction sub-type	"tran_cd"
txn_extra_info	Transaction extra info	"src_in" is source indicator: • 'A' = ATM • 'F' = BFT (Bank for Foreign Trade) • 'T' = Teller • 'U' = Unknown • 'V' = Vault
txn_tm	Time of transaction, in HHmmss format	"tran_tm"

Large cash transactions credit and debit party attributes

There is only one party, either credit or debit. The other party is always null.

Saffron Field Name	Explanation	Notes / Example values
(c|d)r_acc_no	Credit account number	"tran_accno"
(c|d)r_acc_co_no	Credit account company number	"acc_co_no"
(c|d)r_acc_prod_ctgy_cd	Credit account product category code	"acc_prod_ctgy_cd"
(c|d)r_acc_addr	Credit account address	Join with Location table for bank location. Join with "acc_co_no" and "acc_cc_no" to derive the address of the account, if available.
(c|d)r_acc_cty	Credit account city	see above
(c|d)r_acc_st_cd	Credit account state code	see above

(c|d)r_acc_pst_cd	Credit account postal code/zip code	see above
(c|d)r_acc_cntry_cd	Credit account country code	see above
(c|d)r_acc_geocode	Stretch goal feature	
(c|d)r_txn_addr	Credit transaction address	Join with Location table for bank location Join with "**obk_no**" (co_no) and "**orig_cc_no**" to derive the address where the transaction took place.
(c|d)r_txn_cty	Credit transaction city	See above
(c|d)r_txn_st_cd	Credit transaction state code	See above
(c|d)r_txn_pst_cd	Credit transaction postal code/zip code	See above
(c|d)r_txn_cntry_cd	Credit transaction Country code	See above
(c|d)r_txn_geocode	Stretch goal feature	
(c|d)r_customer	From Denormalized data fields shared across schemas	Nested, usually joined via cr_acc_no -> aml_acc_id -> aml_pty_id.

Lookup table for bank location

This is the location lookup table for a customer's branch location.

col_name	data_type
cc_no	int
co_no	int
cc_hier_pt_cd	varchar(10)
hir_alldot_hier_de	varchar(256)
org_hier_itm_1_cd	varchar(256)
org_hier_itm_1_de	varchar(256)
org_hier_itm_2_cd	varchar(256)
org_hier_itm_2_de	varchar(256)
org_hier_itm_3_cd	varchar(256)
org_hier_itm_3_de	varchar(256)
org_hier_itm_4_cd	varchar(256)
org_hier_itm_4_de	varchar(256)
org_hier_itm_5_cd	varchar(256)
org_hier_itm_5_de	varchar(256)
org_hier_itm_6_cd	varchar(256)
org_hier_itm_6_de	varchar(256)
org_hier_itm_7_cd	varchar(256)
org_hier_itm_7_de	varchar(256)
org_hier_itm_8_cd	varchar(256)
org_hier_itm_8_de	varchar(256)
org_hier_itm_9_cd	varchar(256)
org_hier_itm_9_de	varchar(256)

org_hier_itm_10_cd	varchar(256)
org_hier_itm_10_de	varchar(256)
bus_ctr_nm	varchar(50)
addr1	varchar(100)
addr2	varchar(100)
cty_nm	varchar(100)
st_cd	char(3)
zip_cd	varchar(15)
cntry_cd	char(10)
cntry_nm	varchar(50)
rssd_id_no	decimal(10,0)

Large cash transaction metadata

The "txn_metadata" field includes the following:

csh_bx_id
tran_de_tx
appsys_id
tax_id_no
gci_no

Person-to-Person transaction schema

Some definitions:

- "PTS" is Payment and Transfers System
- "Sender Party" is the person who sent the money; that is, debit party.
- "Receiver Party" is the person who receives the money; that is, credit party.
- "Merchant Party" is the merchant through which the external Person-to-Person transactions are carried out, like Facebook, PayPal, or Squarecash.

Person-to-Person common transaction attributes

Saffron Field Name	Explanation	Notes / Example values
isaa_type	ISAA resource type (generated by data ingestion)	"Transaction"
isaa_id	Unique ID used by platform (generated by data ingestion)	"transaction-<txn_type>-<txn_id>-<pst_txn_id>" For example - "transaction-p2p-542854738-43533"
txn_type	Type of transaction	"p2p"
txn_id	ID	"pts_tran_id" - this is P.T.S.
pst_txn_id	Post transaction ID	"pst_tran_id" This is post
txn_am	Value amount of transaction in US dollars This is the primary field the system operates on.	"pts_tran_curr_am"

curr_cd	Currency transaction was executed in	Hardcode "USD": There is a pst_tran_curr_cd field, but it appears to be always null, and I could not find a local currency field
curr_am	Amount in the currency of the transaction	"pts_tran_curr_am": Could not find a separate field for local currency
txn_dt	Date of transaction in "yyyy-MM-dd" format	"pts_src_tran_dt" - format in data is yyyy-MM-dd
pst_txn_dt	Date the transaction posted.	"pst_dt"
txn_sub_type	Transaction sub-type	"pts_src_tran_typ_cd"
txn_extra_info	Transaction extra info	"pts_tran_chn_cd"
txn_tm	Time of transaction, in HHmmss format	"pts_src_tran_tm" - in hhMMss format, no dash, no colon
cr_dr_in	Credit/debit indicator	"tran_cr_dr_in"

Person-to-Person credit and debit party attributes

There is always a "tran party". Credit and debity party is not always in the data. Sometimes there is neither, sometimes of them, and sometimes both.

We'll always use 'tran_accno', since it's always populated. And we'll use 'tran_cr_dr_in'

- If 'tran_cr_dr_in' = 'C', then the credit party info below uses the tran_ party info. The inverse for the 'D' value.
- For the other party that is not specified, the system populates acc_no (if it exists) and the customer party link

Saffron Field Name	Explanation	For the 'chosen' party based on 'cr_dr_in'	Party not chosen
(c|d)r_acc_no	Credit account Number	"tran_accno"	"snd_accno" (if cr_dr_in = 'C', then tran_accno received money, so the other party is the sender) "rcv_accno" (if cr_dr_in = 'D', then tran_accno paid money, so the other party recevied money)
(c|d)r_acc_co_no	Credit account company number	"tran_acc_co_no"	
(c|d)r_acc_prod_ctgy_cd	Credit account product category code	"tran_acc_prod_ctgy_cd"	
(c|d)r_acc_addr	Credit account Address	If 'cr_dr_in' = 'C', then rcv, otherwise, 'cr_dr_in' = 'D', use snd (rcv|snd)_pty_addr_lin_1_tx (rcv|snd)_pty_addr_lin_2_tx (rcv|snd)_pty_addr_lin_3_tx (rcv|snd)_pty_addr_lin_4_tx Data looks null so far, but might as well populate this	The other party in rcv or snd (rcv|snd)_pty_addr_lin_1_tx (rcv|snd)_pty_addr_lin_2_tx (rcv|snd)_pty_addr_lin_3_tx (rcv|snd)_pty_addr_lin_4_tx
(c|d)r_acc_cty	Credit account city	null	null
(c|d)r_acc_st_cd	Credit account state code	null	null
(c|d)r_acc_pst_cd	Credit account postal code/zip code	null	null
(c|d)r_acc_cntry_cd	Credit account country code	null	null
(c|d)r_acc_geocode	Stretch goal feature		
(c|d)r_txn_addr	Credit transaction address	null	null
(c|d)r_txn_cty	Credit transaction city	null	null
(c|d)r_txn_st_cd	Credit transaction state code	null	null

(c|d)r_txn_pst_cd	Credit transaction postal code/zip code	null	null
(c|d)r_txn_cntry_cd	Credit transaction country code	null	null
(c|d)r_txn_geocode	Stretch goal feature		
(c|d)r_customer	From Denormalized data fields shared across schemas	Nested, usually joined via tran_accno / rcv_accno / snd_accno -> aml_acc_id -> aml_pty_id.	

Person-to-Person transaction metadata

tran_pty_email
tran_pty_phn_no
rcv_frst_nm
rcv_lst_nm
rcpnt_nick_nm
snd_frst_nm
snd_lst_nm
rcv_pty_instn_nm
rcv_pty_instn_cd
snd_pty_instn_nm
snd_pty_instn_cd
pts_tran_de_tx
tran_acc_nm
merch_frst_nm
merch_lst_nm
merch_pty_addr_lin_1_tx
merch_pty_addr_lin_2_tx
merch_pty_addr_lin_3_tx
merch_pty_addr_lin_4_tx

Denormalized data fields shared across schemas

These fields are a subset of the customer resource fields. This list of fields is included in the other resource definitions.

Field Name	Nested Field Name	Example Value	Source Table	Notes
AML_PTY_ID		123456789	CPV_PTY_DEMOG_PROFL_CUR	Key value used across multple tables
PTY_ID		12345678912	CPV_PTY_DEMOG_PROFL_CUR	
GCI_Number		null (90%)	CPV_PTY_DEMOG_PROFL_CUR	
Coper_ID		null (90%)	CPV_PTY_DEMOG_PROFL_CUR	
SSN		123456789	CPV_PTY_DEMOG_PROFL_CUR	
Employer_ID_number		null (90%)	CPV_PTY_DEMOG_PROFL_CUR	
MLID		null (90%)	CPV_PTY_DEMOG_PROFL_CUR	

Primary_Party_ID_Number		123456789	CPV_PTY_DEMOG_PROFL_CUR	
Birth_date		1980-01-01	CPV_PTY_DEMOG_PROFL_CUR	
Country_of_Citizenship		US	CPV_PTY_DEMOG_PROFL_CUR	
Country_of_Residence		US	CPV_PTY_DEMOG_PROFL_CUR	
First_Name		Bob	CPV_PTY_DEMOG_PROFL_CUR	
Last_Name		Smith	CPV_PTY_DEMOG_PROFL_CUR	
Email_Address		USA	CPV_PTY_DEMOG_PROFL_CUR	
NAICS_Occupation_Code		null (90%)	CPV_PTY_DEMOG_PROFL_CUR	
Ta_Line_of_Business		ABCDE	CPV_PTY_DEMOG_PROFL_CUR	
Occupation_desc		Labor	CPV_PTY_FIN_PROFL_CUR	
Employer_name		mack's construction	CPV_PTY_FIN_PROFL_CUR	
source_of_income		employment income	CPV_PTY_FIN_PROFL_CUR	
net_worth		null (90%)	CPV_PTY_FIN_PROFL_CUR	
Address				JSON List to show all Addresses
	address_type	physical	CPV_PTY_ADDR_CUR	
	address_line_1	123 happy st	CPV_PTY_ADDR_CUR	
	address_line_2	null (90%)	CPV_PTY_ADDR_CUR	
	city	Cary	CPV_PTY_ADDR_CUR	
	state_code	NC	CPV_PTY_ADDR_CUR	
	state_name	North Carolina	CPV_PTY_ADDR_CUR	
	postal_code	12345	CPV_PTY_ADDR_CUR	
Phone				JSON List to show all phone numbers
	Phone_type	CELL	CPV_PTY_PHN_CUR	
	Phone_Number	1234567890	CPV_PTY_PHN_CUR	

Developing Thought Processes (THOPs)

Note on EA deployment: THOPs were not explicitly exposed to the EA customer. The TPE deployment server was not installed. THOPs for the EA were embedded in the ISAA platform itself.

Thought Processes (THOPs) are JavaScript programs you write after ETL to pinpoint specific data for your precise needs by applying algorithmic logic to that data. For details and methodology on developing THOPs, see Thought Process Engine and THOPs.

ISAA Glossary

The glossary is oriented to *AML* and specific uses of the *ISAA*.

Note: The glossary does not include definitions of many common programming/computing terms, such as HTML, JavaScript, JSON, or R.

- ETL
- FQDN
- geocode
- hypernym
- ingestion
- ISAA
- Know your customer
- lemmatization
- name/value pair
- namelist
- NER
- NLP
- novelty
- origin
- outlier
- path
- pipeline
- processor
- regex
- resource
- Saffron risk score
- SAR
- segment
- signature
- similarity
- space
- stage
- stemming
- Suspicious Activity Report
- THOP
- THOught Process
- TPE
- zone

AML

Anti-Money-Laundering

anomaly

An unusual pattern that does not conform to expected behavior, sometimes also called an *outlier*. Examples of anomalies include:

- Any sudden and substantial increase in funds
- A substantial increase in the velocity (frequency) of transactions
- A large withdrawal
- Moving money to a bank secrecy jurisdiction.
- Smaller transactions that meet certain criteria might also be flagged as suspicious.

Compare *similarity* and *novelty*.

attribute

A value and a *category* with which the value is associated. Each category can be assigned a type as part of a space definition; the type is not stored in the *resource*. The supported types are string (default) and number. Example:

Category	Value	Attribute
`ocean`	`atlantic`	`ocean.atlantic`

An attribute is sometimes called an "entity".

Compare the programming construct *name/value pair*.

Bank Secrecy Act

US law for combating money laundering and terrorist financing. Codified in Title 31 USC 5311.

BSA

See *Bank Secrecy Act*.

category

A classification of a value. The left hand side of a *name/value pair*. Sometimes a category is a *hypernym*. See also *attribute*.

CDH

Cloudera open source big data software with integrated Apache Hadoop

data drift

A common phenomenon in a machine learning or other AI systems: data changes over time, requiring re-evaluation and perhaps redesign or reprogramming.

destination

StreamSets term for where data that has been transformed via *processors* is sent. The end of a *pipeline*, the sink for ouput from ETL. See also *origin*.

dimension

An ordered relationship in a data continuum, such as time or physical space. A secondary aspect that modifies or constrains another datum. Typically described with the word "by", as in "transactions **by time**" or "outgoing transfers **by location**".

distance

The result of a calculation of the *similarity* between two or more objects. Some kinds of distance are:

- inherent, such as with time or numbers
- geographical distance-based
- feature-based
- psychological

entity

Synonym for *attribute*.

ETL

"Extract, Transform, Load." A process in computing for pulling data out of source systems, changing the data, and making it available to other systems (sometimes by placing it into a data warehouse).

FQDN

Fully qualified domain name of an Internet-connected computer

geocode

Formal notation for the longitude and latitude of a location on the surface of the Earth.

Geocode information is supplied by the GeoNames postal and city downloads available under the Creative Commons Attribution 4.0 License.

hypernym

A word with a broad meaning that more specific words fall under. A superordinate. For example, "color" is a hypernym for the following:

- red
- green
- blue

ingestion

Transferring data from one system to another, usually transforming it for use in the new system. See also *ETL*.

ISAA

Intel Saffron AML Advisor

Know your customer

A key goal of AML involving analysis of patterns of customer behavior to establish common financial characteristics about that customer, such the kinds of transactions in which the customer is likely to engage. By knowing one's customers, financial institutions can often identify unusual or suspicious behavior, termed *anomalies*, which may be an indication of money laundering.

lemmatization

Part of *NLP*, a subtask for processing text with the use of a vocabulary and morphological analysis of words. See also *stemming*.

Lemmatization, like stemming, tries to group related words, but it goes farther than stemming in that it tries to resolve ambiguity by grouping words by their word sense, or meaning, not by their specific grammatical form. The same word may represent two meanings—for example, “wake” can mean "to wake up" or a “funeral”.

name/value pair

In programming, a data structure that assigns a value to a variable. The name of the variable is similar to a classification or *category* for the value.

The left-hand-side is the name. The right-hand side is the value.

The name is often a *hypernym*, a superordinate of the value.

Arrays of name/value pairs are often combined to form a *namelist*, which is useful in *Named Entity Recognition*.

namelist

Programming construct for input or output of whole groups of variables, or input of selected items in a group of variables, usually in the form of an array. It specifies a group name to list the variables and arrays belonging to that group.

NER

Named Entity Recognition. Part of *NLP*, a subtask of information extraction that seeks to locate and classify named entities in text into pre-defined *categories* such as the names of persons, organizations, locations, expressions of times, and so on.

NLP

Natural Language Processing. Some terms in NLP include:

- *hypernym*
- *lemmatization*
- *NER*
- *stemming*

novelty

A previously unnoticed observatiion of a pattern in the data not originally included or accounted for by *processors*. Distinct from *anomaly*. The novel pattern is typically added back to the data transform processors to account for the previously unobserved pattern and thus remove the novelty. Compare *similarity* and *anomaly*.

origin

StreamSets term for where particular input data comes from, a data source. The start of a *pipeline,* which ends in a *destination*.

outlier

Synonym for *anomaly*.

path

In machine learning, a probability path is designed for humans who require a deep understanding of advanced probability for their research or applied use in statistics, biology, operations research, mathematical finance (such as *AML*), engineering, and other disciplines.

In topology, a path is a continuous mapping, with an initial point, a final point, and the space of continuous functions between them. In a topological space *X*, a path is a continuous function *f* from the unit interval *I* = [0,1] to *X*. *f* : *I* ? *X*. The initial point of the path is *f*(0) and the terminal point is *f*(1).

In graph theory, a path in a graph is a finite or infinite sequence of edges which connect a sequence of vertices which, by most definitions, are all distinct from one another.

See also *signature*.

pipeline

StreamSets term for a communications/transformation channel for incoming data. With an *origin* and a *destination*, a pipeline includes discrete *stages* that run *processors* to perform a particular change (or "transformation") on the incoming data.

processor

A defined programatic function that transforms incoming data, included as a *stage* in a *pipeline*. From StreamSets.

regex

Regular expression, a text pattern matching mask. See https://en.wikipedia.org/wiki/Regular_expression.

resource

A collection of *attributes* and optional structural information. The *origin* of data for a *pipeline*.

Saffron risk score

A measure of the probability of risk based on the distance from the established pattern of customer behavior, based on specific attributes. See Metrics and Scores.

SAR

See *Suspicious Activity Report*.

segment

An ordered list of *attributes* or other segments. Segments are identified by a label, which is a string.

signature

A mathematical expression that quantifies a *path*, an evolving or time-ordered sequence of events, parameterized by a continuous variable.

similarity

The state of “likeness” between two or more objects expressed by a mathematical formula. The formula is a quantification of the degree of

similarity, which is called *distance*. See also *anomaly* and *novelty*.

space

In a Docker multi-tenancy deployment, a *zone* containing spaces is a segregated area for protecting and isolating processes and data for specific purposes and specific groups of users.

In analogy with a physical apartment building with many tenants, a zone is a single, locked apartment. The zone/apartment is further subdivided into individual rooms, one per person (or group of users). The rooms an analogy for Docker spaces, which protect data specific to that group of users.

stage

A discrete, identified portion of a *pipeline* where *processors* transform incoming data. From StreamSets.

stemming

In linguistic morphology and information retrieval, stemming is the process of reducing inflected (or sometimes derived) words to their word base or root form, which is generally a written word form. Example: "send" is the stem of:

- send
- sending
- sent

See also *lemmatization*.

Suspicious Activity Report

After a suspected incident of money laundering or fraud, financial institutions must file a SAR report with the Financial Crimes Enforcement Network (FinCEN) of the US government. These reports are required by the United States Bank Secrecy Act (BSA) of 1970.

THOP

THOught Process. A JavaScript program you write for computing results from a Saffron memory store, relying on algorithms you implement to produce meaningful results. These THOPs are packaged into a library you create and load into the *TPE* deployment service for use with the *AML* Advisor.

THOught Process

See THOP.

TPE

Thought Process Engine. *ISAA*'s computing service that processes *THOPs*.

zone

In a Docker multi-tenancy deployment, a zone is a segregated area for protecting and isolating processes and data for specific purposes and specific groups of users.

In analogy with a physical apartment building with many tenants, a zone is a single, locked apartment. The zone/apartment is further subdivided into individual rooms, one per person (or group of users). The rooms an analogy for Docker *spaces,* which protect data specific to that group of users.

Revision history: Draft: ISAA AML Architect Guide, EA edition

Date	Description
2018-01-30	Inspection session

2017-12 – 2018-01	Working drafts for internal reviews