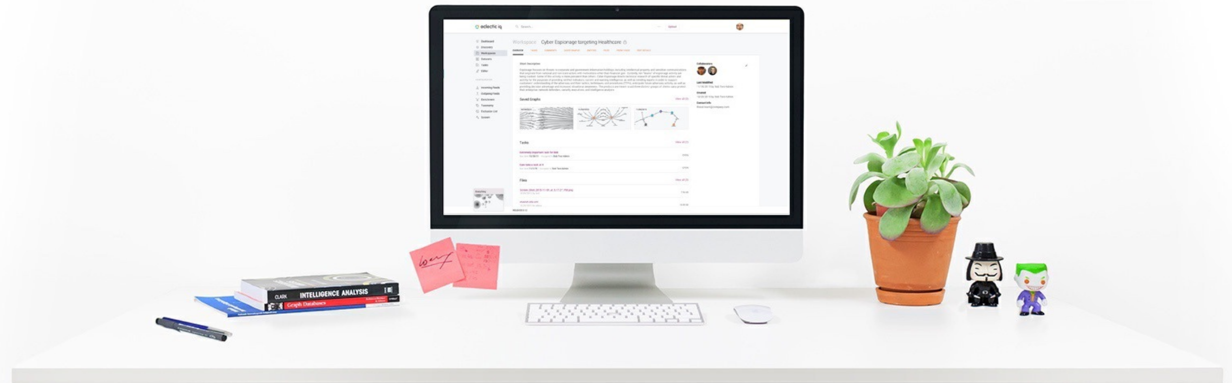
How-tos for EclecticIQ Platform

Hands-on articles on specific platform features

Last generated: March 06, 2018



Last generated on Mar 6, 2018

Table of contents

How to split MISP STIX packages

Split MISP STIX packages into their constituent embedded STIX packages by using the MISP splitter command line utility.

Issue

MISP XML files usually include multiple STIX XML packages. Each embedded STIX package holds data defining an entity object. The parent MISP XML file serves as a container for the embedded STIX packages. When a MISP XML file holds a large number of STIX packages, it may cause ingestion errors.

To address this issue and to correctly ingest all the valid STIX content, you can split the source MISP XML package into its constituent embedded STIX packages. This process removes the MISP XML container layer, and it outputs one XML file per STIX XML sub-package. The platform can then ingest and process the resulting STIX XML packages.

Solution

The EclecticIQ Platform ships with a command line utility that splits the embedded STIX packages into separate XML files: *split-misp-stix-packages*.

```
#!/usr/bin/env python

import logging
import os
import sys

import click

import objectivistix

logger = logging.getLogger('')


@click.command()

@click.option(
    '--output-directory',
    type=click.Path(
        exists=True,
        file_okay=False,
        dir_okay=True,
        writable=True),
    required=True,
    help="Output directory")

@click.option(
    '--output-base-name', default='package',
    help="File name template to be used for output files")

@click.option('--debug', is_flag=True)
```

```
@click.argument(
    'input_file', metavar='FILE.xml', type=click.File(mode='rb'))
def main(output_directory, output_base_name, debug, input_file):
    logging.basicConfig(
        format='%(asctime)s [%(levelname)s] %(message)s',
        level=logging.INFO)
    logging.getLogger('objectivistix').setLevel(logging.ERROR)

    logger.info("Loading file %s", input_file.name)
    root_pkg = objectivistix.obj_from_xml_file(input_file)
    root_pkg.setdefault('id_namespaces', {})

    # MISP stores real content as embedded 'related packages'.
    logger.info("Splitting embedded 'related packages' into separate files")

    n = 0
    for n, pkg in enumerate(root_pkg.get('related_packages', []), 1):
        pkg['id_namespaces'] = root_pkg['id_namespaces']

        path = os.path.join(
            output_directory,
            '{}-{:06d}.xml'.format(output_base_name, n))
        with open(path, 'wb') as fp:
            logger.info("Creating %s", path)
            objectivistix.obj_to_xml_file(pkg, fp)

    logger.info("%d output files written", n)


if __name__ == '__main__':
    try:
        main()
    except Exception as exc:
        logger.error("%s: %s", type(exc).__name__, exc)
        if '--debug' in sys.argv:
            raise
        sys.exit(1)
```

To run the script correctly, follow these recommendations:

- Run *split-misp-stix-packages* inside the CentOS virtual environment the platform runs on. Currently supported: **CentOS Linux 7 (1511)** `(https://lists.centos.org/pipermail/centos-announce/2015-december/021518.html)`.
- Explicitly point to the Python interpreter inside the virtual environment.
  Example:

```
# Python interpreter included with the
# virtual environment the platform runs on.
/opt/eclecticiq/platform/api/bin/python
```

Usage

- Activate a Python virtual environment:

```
$ source /opt/eclecticiq/platform/api/bin/activate
```

- To access the *split-misp-stix-packages* script do the following:

```
(api) [root@IP bin]# /opt/eclecticiq/platform/api/bin/python split-misp-stix-packages
```

- To view the built-in help, run the following command(s):

```
# Enter this command:
$ /opt/eclecticiq/platform/api/bin/python split-misp-stix-packages --help

# The help is displayed:
Usage: split-misp-stix-packages [OPTIONS] FILE.xml

Options:
  --output-directory DIRECTORY  Output directory  [required]
  --output-base-name TEXT       File name template to be used for output files
  --debug
  --help                        Show this message and exit.
```

Option	Description
`--output-directory`	*Required* — Defines the file splitter output directory the embedded STIX packages are saved to after extracting them from the MISP XML wrapper.
`--output-base-name`	*Optional* — By default, STIX packages are named `package-${int}.xml`, where `${int}` is a sequentially progressive numeric value starting at zero. If you want, you can specify a different name than `package`. It is not possible to modify the hyphen or the numeric part of the file name.
`--debug`	*Optional* — In case of errors, you can use this option to return a verbose output.
`--help`	*Optional* — Displays the built-in help.

Example

In this example, we are using the following dummy names for files and directories:

- `${platform_virtual_environment_username}`: user name to access the CentOS virtual environment the platform runs on. Currently supported: **CentOS Linux 7 (1511)** `(https://lists.centos.org/pipermail/centos-announce/2015-december/021518.html).`
- `${platform_virtual_environment_password}`: password to access the CentOS virtual environment the platform runs on. Currently supported: **CentOS Linux 7 (1511)** `(https://lists.centos.org/pipermail/centos-announce/2015-december/021518.html).`
- `${temp}`: directory where the source `${misp-out misp-stix-package.xml}` file is temporarily stored.
- `${misp-stix-package.xml}`: example MISP XML file to extract the embedded STIX packages from.
- `${misp-out}`: output directory where the embedded STIX packages, originally in the MISP XML file, are saved as separate XML files.

These are the main steps:

- To successfully execute several commands in the command line or in the terminal, you may need root-level access rights.
  To obtain admin rights, run the following command(s):

```
$ sudo su -
```

Alternatively:

- Grant admin rights to a specific user, who can then log in with their password to perform admin tasks:

```
$ su - ${username}
```

Or:

- Prefix `sudo` to the command you want to run:

```
$ sudo ${command}
```

- Create an output directory where the STIX packages can be saved as separate XML files.
- Go to the directory where the MISP XML file you want to split is located.
- Run the file splitter utility.

```
# SSH authentication in the EclecticIQ Platform virtual environment.
$ sudo su ${platform_virtual_environment_username}
password: ${platform_virtual_environment_password}

# Create a new directory; the MISP XML sub-packages will be saved here.
$ mkdir misp-out

# Go to the directory where you saved to source MISP STIX XML package.
# Example: "temp".
$ cd temp

# Run the split utility tool. Specify:
# - The output directory for the split sub-packages.
# - The source MISP STIX XML package you want to split.
$ /opt/eclecticiq/platform/api/bin/python split-misp-stix-packages --output-directory misp-out
misp-stix-package.xml

# Log message at the end of a successful operation,
# where "%d" is an integer.
%d output files written
```

How to create a money mule TTP

Create a money mule TTP entity to investigate fraudulent activities and to identify the actors involved in them.

Money mules are middlemen who carry out illegal transactions on behalf of a criminal third party. Money mules may not always be aware that they are engaging in criminal activities aimed at committing fraud. They are part of a larger scheme designed to carry out fraudulent transactions involving money or goods.

In a fraudulent financial transaction, money mules are responsible for laundering the illicitly obtained money such as proceeds from phishing, malware or email scams. They transfer the money using money orders or cryptocurrencies, which provide an effective layer of obfuscation.

To identify and to track these actors and their behavioral patterns, fraud and risk teams can create TTP entities that describe the actors, their behaviors, and the victims. Analysts can add relationships with other entities on the fly, as well as let the platform process the data to generate meaningful intelligence providing valuable context to their investigation.

In the EclecticIQ Platform, you always record a money mule as a TTP entity where you need to include at least:

- An actor (the money mule).
  The TTP entity describes the money mule as a malicious actor by defining the context the money mule operates in as accurately as possible.
- A victim (for example, a bank account).
  You define and describe the victim of a money mule in the **Characteristics > Targeted Victim** section.
  A victim can be an individual, a commercial or financial entity, or an object like an email address.
- An intended effect of the criminal behavior (for example, fraud).
  You select the intended effect a money mule aims to achieve in the **Intended effects** section.
  Such an effect can be fraud, theft, money laundering, and so on.

Create a money mule TTP

To create a TTP entity describing a money mule, do the following:

- On the left-hand navigation sidebar, click **Editor**.
- On the editor page, click the ✚ **Entity** button.
- From the drop-down menu select **TTP**.

The entity editor is displayed, and you can proceed to create the new TTP entity.

✓ Input fields marked with an asterisk are required.

Title

Specify the name of the new entity. It should be descriptive and easy to remember.
For example: *Money mule related to IBAN ${bank_account_number}*

Analysis

it is a free-text input field to include non-structured information such as additional context, references, links, and so on.
For example, you can add contextual details that can help identify the money mule or the location they operate in.

Confidence

From the drop-down menu select an option to assign the entity a confidence value.
it flags the **estimated level of confidence** `(https://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part14-vocabularies/stix-v1.2.1-csprd01-part14-vocabularies.html#_toc440440605)` to assess the accuracy and trustworthiness of the entity information.

Intended effects

From the drop-down menu select an option to specify the purpose or the goal the cyber threat aims at achieving.
Fraud is a very common effect money mules and their associates intend to achieve.

Characteristics

This field allows you to add extra details to more accurately describe the entity; for example, by specifying the threat type, the resources it uses to spread and to reach the intended target, or any connections with other entities.
The one characteristic you want to include in a money mule TTP entity is **Targeted Victim**.

Create a targeted victim

Use the **Characteristics > Targeted Victim** section to record information about the individual, the organization, or the resources affected by the money mule's behavior:

- Under **Characteristics**, click **✚ Characteristic**, and then select **Targeted Victim**.
- The **Targeted Victim** editor opens. It is based on the **CIQ standard** `(https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ciq)` and its **specifications** `(http://docs.oasis-open.org/ciq/v3.0/specs/ciq-specs-v3.html)`. The Customer Information Quality specification aims at providing an open and standard data model to accurately and consistently describe a party such as an individual or an organization, as well as attributes like roles and relationships.
  Apart from drop-down menus and checkboxes, where available, the editor input fields accept free-text as an input.
  No field is mandatory.

Name: specify the name of the targeted victim. It should be descriptive and easy to remember.
Example: *IBAN ${ludicrously_fat_bank_account_number}*

Under **✚ Characteristic > Targeted victim > Specification** you can define the type of victim under attack. You can describe affected individuals, organizations, and assets.

- Click **✚ Fields**.
  From the drop-down menu select an option to define the type of targeted victim:
  - **Account**
  - **Person**
  - **Organization**
  - **Electronic address**

Targeted systems: from the drop-down menu select **one or more entries** `(https://stixproject.github.io/data-model/1.2/stixvocabs/systemtypevocab-1.0/)`, as applicable, to describe the type of infrastructure, system or equipment affected by the threat actor's TTP.
Example: *Enterprise Systems — Database Layer*

Targeted information: from the drop-down menu select **one or more entries** `(https://stixproject.github.io/data-model/1.2/stixvocabs/informationtypevocab-1.0/)`, as applicable, to describe the type of information being handles or manipulated in the TTP.
Example: *Information Assets — Financial Data*

Specify the targeted victim type

- Under **Characteristics > Targeted Victim > Specification**, click **✚ Fields**.
  The available types allow you to describe affected individuals, organizations, and assets.

Account

Account type: defines the type of account related to the victim.
Example: *bank*, *online*

Account status: defines the current status of the account.
Example: *active*, *blocked*

Account specification: this section takes a set of predefined keys you can select from the drop-down menu, along with the corresponding values you enter as free-text in the input fields.

Click **✚ Add** or **✚ More** to insert a new empty row below the current one, which you can populate with additional details.

Key	Value
Account ID	The account number. Example: *NL30INGB0123456789*
Issuing Authority	The financial institution that issues the account. Example: *ABC Bank*
Account Type	The type of account. Example: *debit* or *savings*
Account Branch	The local branch office or the retail location of the bank responsible for issuing the account. Example: *Utrecht center*
Issuing Country Name	The name of country where the account was issued. Example: *The Netherlands*

Person

Person name: this section takes a set of predefined keys you can select from the drop-down menu, along with the corresponding values you enter as free-text in the input fields.

Click **✚ Add** or **✚ More** to insert a new empty row below the current one, which you can populate with additional details.

Key	Value
Preceding Title	Example: *His*, *Her*
Title	Example: *Rogueness*, *Excellence*, *Pandit*, *Sheikh*
First Name	Example: *Peter*
Middle Name	Example: *Brandon*
Last Name	Example: *Quill*
OtherName Name	Example: *Guardian of the Galaxy*
Alias Name	Example: *Star-Lord*

Key	Value
Generation Identifier	Example: *Jr.*, *Sr.*, *The Younger*, *The Elder*, *XXVIII*
Degree	Example: *BSc Ethical Hacking*

Organization

Organization name: this section takes a set of predefined keys you can select from the drop-down menu, along with the corresponding values you enter as free-text in the input fields.

Click ✚ **Add** or ✚ **More** to insert a new empty row below the current one, which you can populate with additional details.

Key	Value
Name Only	The name the organization is commonly referred to. Example: *Wey-Yu*
Type Only	The entity definition of the organization. Example: *Inc*, *LLC*, *Ltd*
Full Name	The full name of the organization. Example: *Weyland-Yutani Corporation, Inc.*

Electronic address

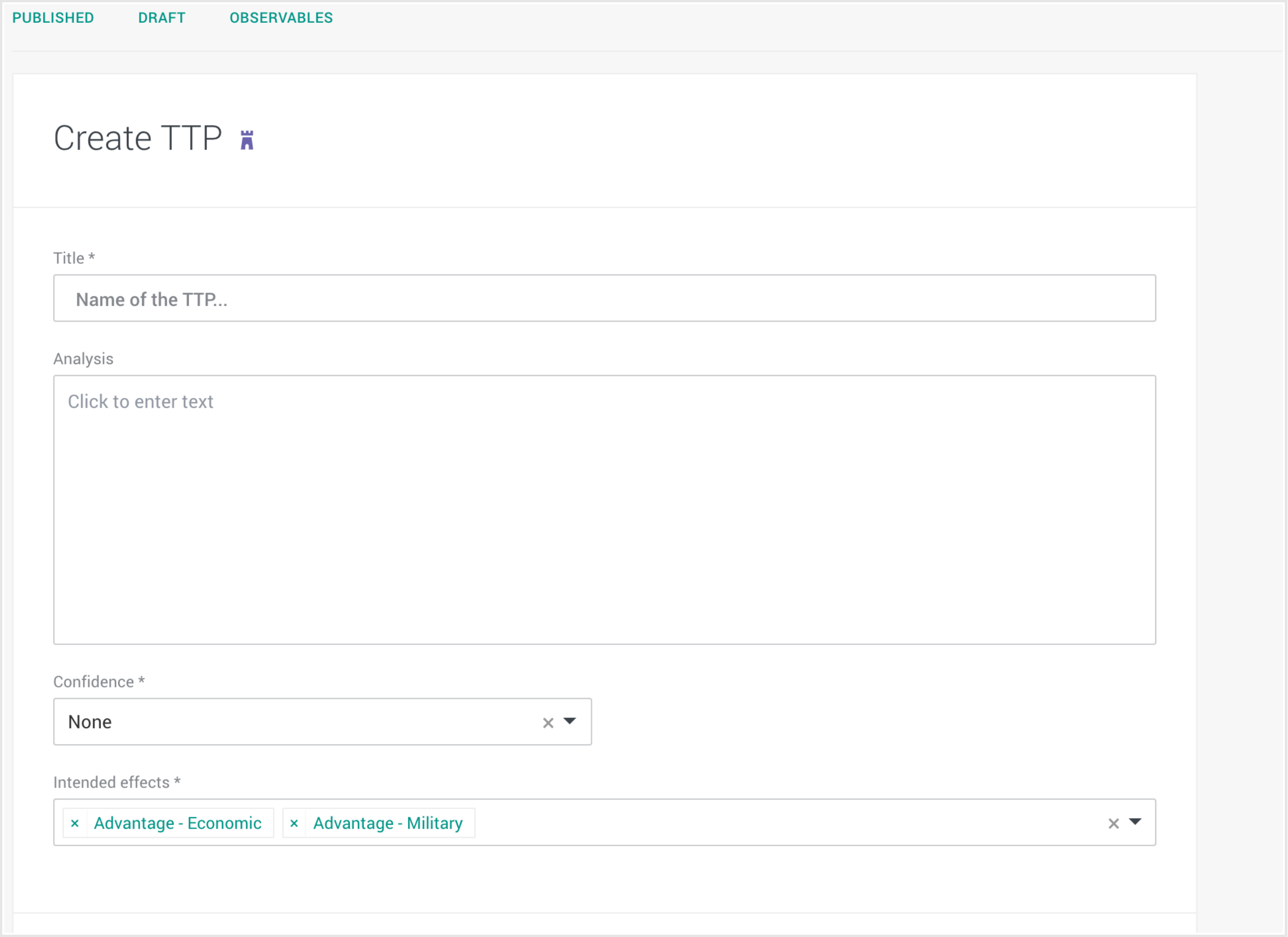
Electronic address: this section takes a set of predefined keys you can select from the drop-down menu, along with the corresponding values you enter as free-text in the input fields.

- The key corresponds to the service provider, for example Google, Yahoo, Skype, ICQ, and so on.
- The associated value needs to be a valid format for the selected service provider, for example:
    - Google: *larry@gmail.com*
    - Yahoo: *melinda-ex@yahoo.com*
    - Skype: ${skype_username}*

Next steps

To complete the money mule TTP entity creation, follow the standard steps and procedures you normally use to create entities in the editor, tag them, add relationships, and enrich them with observables.

Example



Create TTP

Title *

Name of the TTP...

Analysis

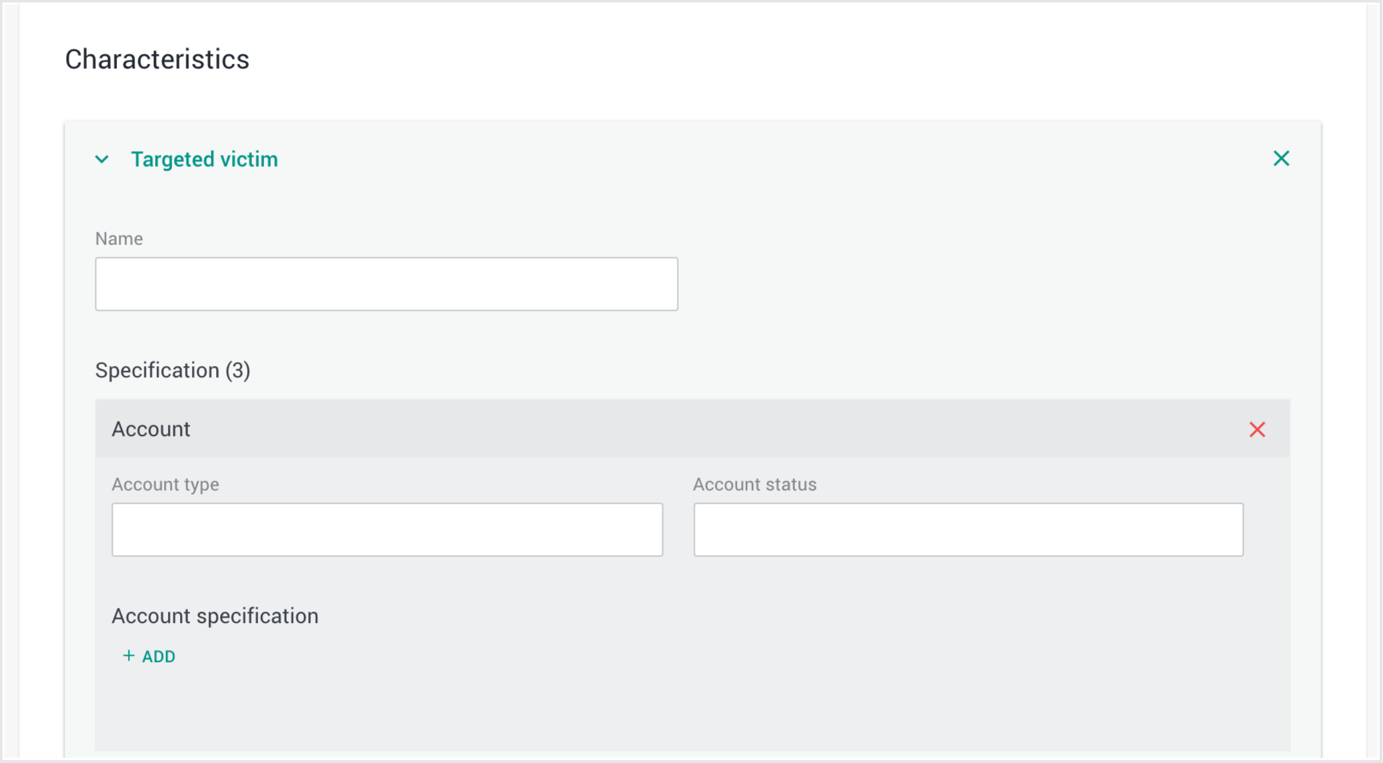
Click to enter text

Confidence *

None

Intended effects *

× Advantage - Economic × Advantage - Military



Characteristics

Targeted victim

Name

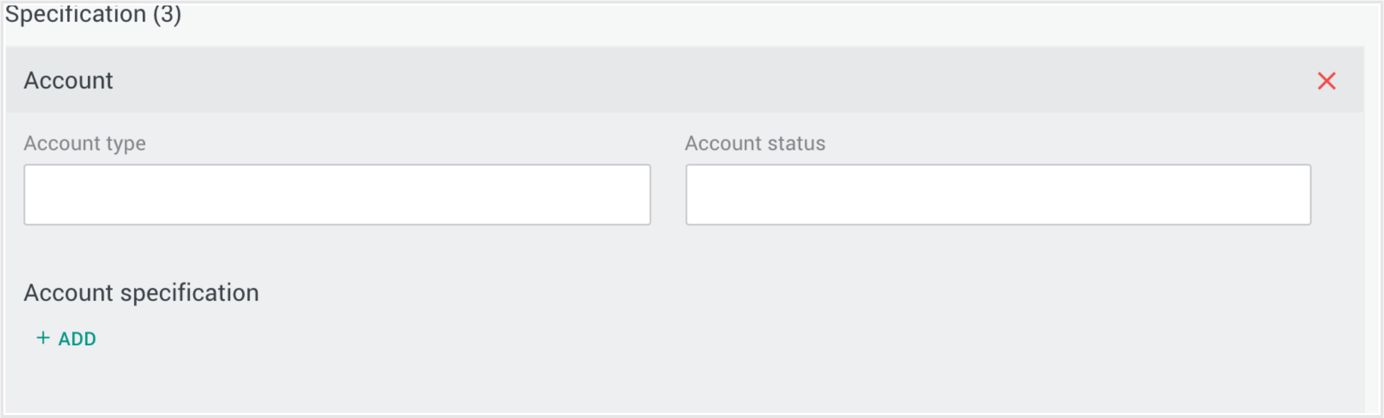
Specification (3)

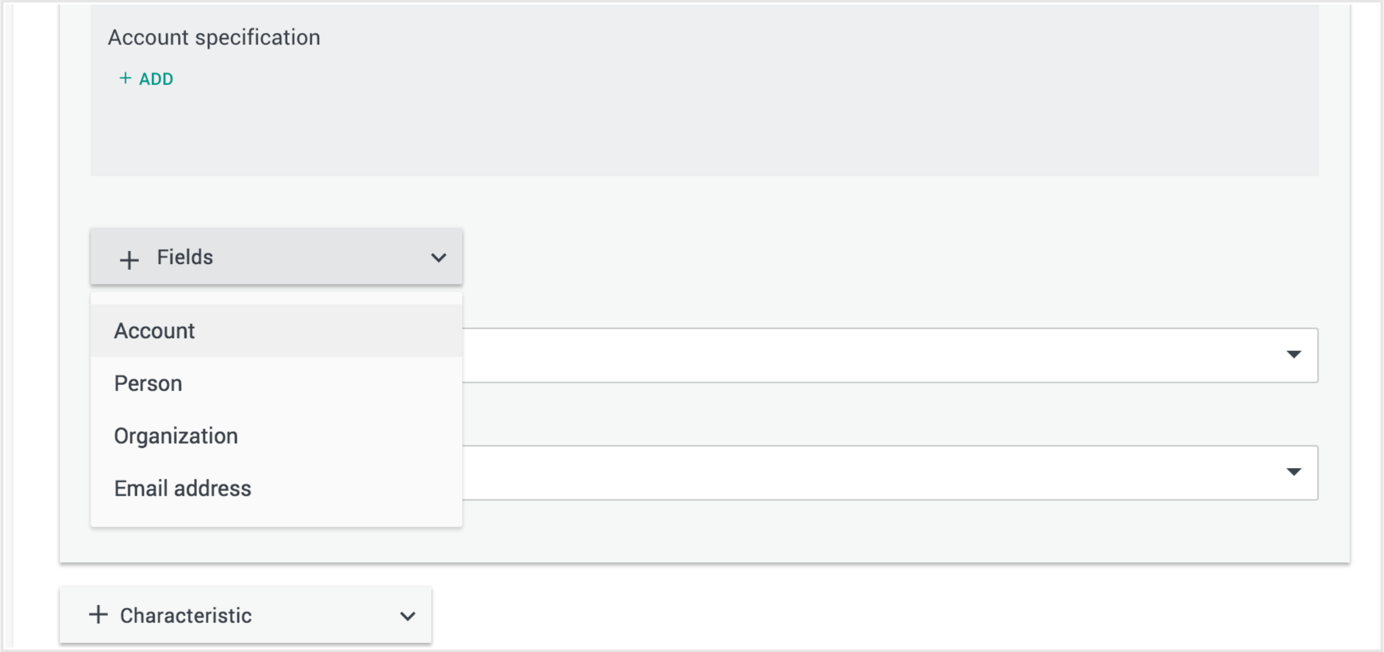
Account

Account type

Account status

Account specification


Specification (3)
Account
Account type
Account status
Account specification
+ ADD


Account specification
+ ADD
+ Fields
Account
Person
Organization
Email address
+ Characteristic