Standard Elements Importer

SEI 1.37

How to import category data into Elements.

Document Conventions

As some terms used in this documentation have a specific meaning when used in relation to Elements they will appear in italics to indicate that the meaning is as it is understood within Elements and not its literal meaning.

***This document refers to SEI 1.37 .If you are using an earlier version, some* features *may not be available.*

Glossary

SEI: Standard Elements Importer

Category: Import type. Appendix A defines a list of allowable import types.

Type: Categories contain types. Appendix A defines a list of allowable category types.

Field: Record fields that hold the metadata (appear as column headers in the SEI input files).

Default Types: All the native field types that are defined in Elements and are distinct from Customised types defined by clients.

Customized Field Types: Defined by a user to capture metadata that can not be mapped to any existing default fields.

Underlying Field Type: Both default and custom fields have an underlying field type associated with it. Appendix B defines the list of allowable field types. This list is definitive and cannot be extended.

Contents

How the SEI works

The Standard Elements Importer (import tool) is a utility that can be used to import data from formatted CSVs into Elements via the Elements API.

The tool will import records of a specific category, which may result in a new object in Elements with a single record(the imported one) or for publications, grants or journals, these records may be merged into existing objects.

IMPORTANT: SEI is just creating a simple records, Elements will decide if the record should merge with an existing object or it should create a new object for the imported record.

It is important to decide what source to use when importing records data. It is recommended to use the manual source for one time data migrations(were you may want to be able to edit the record manually after the migration) and the institutional sources for ongoing data feed/ pipelines.

IMPORTANT: For ongoing data feeds/pipelines it is mandatory to configured the importer to import the records with an institutional source. If we start with the manual source, in order to change later to an institutional source, we need to delete all imported records previously with the manual source.

The import tool will attempt to parse data to conform to the relevant Elements **field types**. All **categories** are supported, and objects of any **type** (including custom ones), currently configured in Elements can be imported.

As a high-level overview, an import consists on the following steps:

1. **Prepare CSV files**: Produce CSVs in the format defined here. Place them in the directory defined in the import tool section. The tool comes with a set of sample files that can help you with this process.
2. **Configure import tool**: Update the CONFIG file to set the correct file locations, the API details for the Elements API endpoint and the SMTP settings for emailing error reports.
3. **Run import tool**: Run the import tool in test mode with a few IDs to verify that it runs successfully. Then, run the full import

The complete import for a particular item happens in two steps:

1. SEI processes the metadata CSV row by row, and for each row, it searches for an entry with the same ID in the persons CSV to collect additional data for "person" or "person-list" metadata field types. Using the data from these two files, it initiates a PUT API command to create a record with the specified type and category. In this step, only the record is created, without any relationship with other objects.
2. If this initial API call succeeds, it uses the same ID from the current row in the metadata file to find relationship details in the links CSV file, triggering a second API call to create all the necessary relationships.

IMPORTANT: To start the process, you will need to map the source data to Elements data structure. This mapping will help you with the generation of the needed CSV input files, where Elements category, types and field names have to be used in header names or as values. Mapping templates for all categories can be found here:

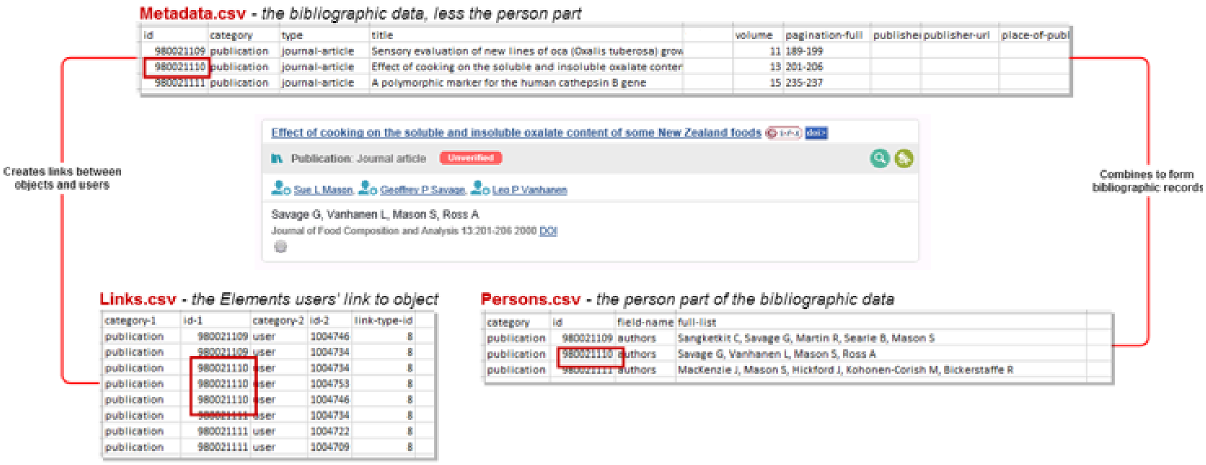
🟩 [Client] - Publications Mapping File 🟩 [Client] - Teaching Activities Mapping File
🟩 [Client] - Grants Mapping File 🟩 [Client] - ProfessionalActivity Mapping File

1. Prepare CSV files

The importer accepts data from three separate CSV files:

- **Metadata.csv**
- **Persons.csv**
- **links.csv**

How the CSV files relate to each other


Creates links between objects and users
Combines to form bibliographic records

Metadata.csv - *the bibliographic data, less the person part*

id	category	type	title		volume	pagination-full	publishe	publisher-url	place-of-publ
980021109	publication	journal-article	Sensory evaluation of new lines of oca (Oxalis tuberosa) grow		11	189-199			
980021110	publication	journal-article	Effect of cooking on the soluble and insoluble oxalate conter		13	201-206			
980021111	publication	journal-article	A polymorphic marker for the human cathepsin B gene		15	235-237			

Effect of cooking on the soluble and insoluble oxalate content of some New Zealand foods

Publication: Journal article Unverified

Sue L Mason, Geoffrey P Savage, Leo P Vanhanen

Savage G, Vanhanen L, Mason S, Ross A
Journal of Food Composition and Analysis 13:201-206 2000 DOI

Links.csv - *the Elements users' link to object*

category-1	id-1	category-2	id-2	link-type-id
publication	980021109	user	1004746	8
publication	980021109	user	1004734	8
publication	980021110	user	1004734	8
publication	980021110	user	1004753	8
publication	980021110	user	1004746	8
publication	980021111	user	1004734	8
publication	980021111	user	1004722	8
publication	980021111	user	1004709	8

Persons.csv - *the person part of the bibliographic data*

category	id	field-name	full-list
publication	980021109	authors	Sangketkit C, Savage G, Martin R, Searle B, Mason S
publication	980021110	authors	Savage G, Vanhanen L, Mason S, Ross A
publication	980021111	authors	MacKenzie J, Mason S, Hickford J, Kohonen-Corish M, Bickerstaffe R

The primary table is the **metadata.csv** file as it contains the primary key that joins the persons and links tables together so the metadata **id** field is the foreign key in the other two tables.

- metadata.id -> persons.id
- metadata.id -> links.id-1 (or links.id-2 depending on the link type ID used)

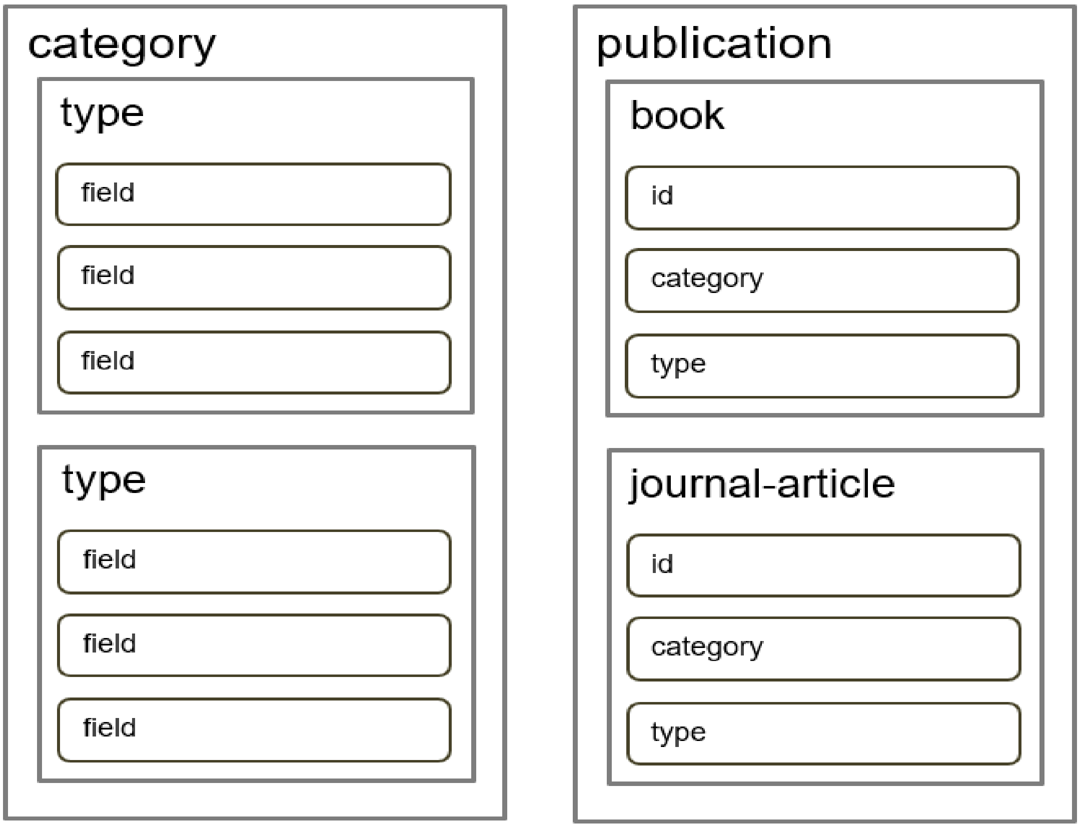
Metadata CSV

It is used to create records with an unique **id(**record proprietary id / id at source **)**, under a specific **category**, with specific **type** and metadata fields.

The metadata.csv file contains the three main components required by the Importer and they are:

- **ID:** Each row in the metadata should have a unique ID per category. This id represents the record's proprietary ID, the ID of the record on the source system, an ID that you can reference in the future to be able to update the record if needed.
- **Category:** The records that are to be created need to target an existing Elements category such as Publications (See Appendix A for complete set). Please use the identifier names, not display names.
- **Type:** Each *category* can contain a number of *types* so a Publication could be of *type* book, chapter, or journal-article (See Appendix A for each category and their valid types.)Please use the identifier names, not display names.
- **Any additional Elements metadata field:** Each *type* has a set of metadata *fields* that contain the metadata associated with a specific *type*. To populate specific metadata fields in the records that are about to be imported, we need to use the underlying field names as headers(lowercase). (See Appendix C for complete list)

The following diagram illustrates the *Elements* import object model and how the individual components relate to each other and how a component maps to actual values within the CSV file.



A *category* can contain a number of different *types* and each *type* has metadata associated with it in the form of *fields*. This is modeled in the metadata.csv file by using column headings that contain defined values or free form values depending on the data-type of the fields. These heading must be either from the list of default fields native to *Elements* or * custom fields that you have created, for which the underlying field name will always start with "c-" for example "c-location". (See Appendix C for complete default list).

TIP: The additional header names in the metadata file must correspond precisely to the underlying field name of the target field. For instance, to populate the "Start Date" field, the header name for the CSV column containing the start date values must be 'start-date,' which is the underlying field name. There are a few exceptions where the headers for a specific field are not an exact match with the underlying field names. For example, to populate pagination or money field types, we need special headers.

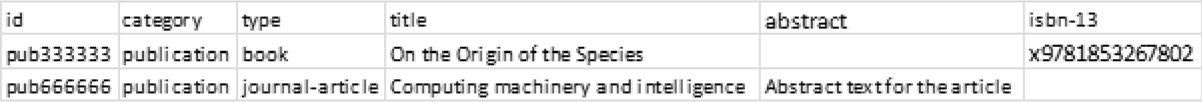
During its operation, the SEI generates a *Elements_Type\Configuration\\[date\].csv* file within the Logs directory. This file includes all the *category/type/field* configurations present in that specific instance of *Elements*. This is how the SEI determines which header names to recognize. Header names containing spaces, such as "Start date," will cause the tool to malfunction, while any other header names will be disregarded.

This export can also be helpful for debugging purposes, as it provides a precise description of the information contained within that particular instance.

The importer will throw an exception if any of the three mandatory fields in the metadata.csv file are missing a value.

- **id:** The primary key**.** This constraint is not enforced as part of the import so validating that it is unique must take place prior to importing the file. The values in this column should represent unique identifiers for each record. These values will become the record's proprietary ID in Elements.
- **category:** A column must contain a defined *category* name. (See column A in Appendix A for the full list)
- **type:** A column must contain a defined *type* name under the targeted category. (See Appendix A for the full list)

An **important point to highlight** here is that metadata.csv contains all the *fields* for all the *category types* that are to be imported. So if the file is importing books and journal-articles the file will contain fields that are only relevant to a book and not a journal-article and visa-versa. They will all be present as column headings but only filled in if they relate to the relevant *type*.



id	category	type	title	abstract	isbn-13
pub333333	publication	book	On the Origin of the Species		x9781853267802
pub666666	publication	journal-article	Computing machinery and intelligence	Abstract text for the article	

This example demonstrates how the table contains two publications of different types: 'book' and 'journal-article' the abstract field is only relevant to the article so it's blank for the book and filled in for the article. The book has the isbn-13 field in but the article doesn't.

You may notice in the sample files that the metadata.csv has many fields but this does not imply that they all have to be present in every import. Only fields that are needed to map your metadata need to be added as column headings.

So the example above would be a valid metadata.csv file as it has the three required fields and additional default fields 'abstract' and 'isbn-13'. As a real-world file for articles would have an ISSN this would be added as a column heading and then populated with the article ISSNs.

TIP If you have a piece of metadata to import add the appropriate field as a column heading. If you don't have data that maps to a field there is no need to add the column heading.

TIP Even if the fields of type person or person-list are simply metadata field for the import context, these are not part of the metadata file; they are populated using the Persons.csv file.

TIP The syntax of the column headings must be as defined in Appendix C

Persons CSV

The persons.CSV file is used to populate metadata fields of type ‘person’ or ‘person-list’ in the first step when the record is created alongside with the data from the metadata.csv. These fields may contain names of internal and external persons, the main purpose of these fields is to store the person names, but Elements may suggest pending relationships to existing users if the names in fields like authors match the search settings of an existing user. In the extended version of persons file, we can add also the ORCID or email for a named persons, this will also help Elements to create claimed relationships. The import tool is simply populating these metadata fields, nothing more, Elements may resolve these names.

The persons.csv file has three required fields that must contain a value or the importer will not be able to import the metadata.

- **id:** This is the id from the metadata.csv for the record of that item i.e. publication and has no constraint to be unique.
- **category:** A column must contain a defined *category* name. (See Appendix A for the full list.)
- **field-name:** Holds the relationship of the person to the metadata record

This file will not create relationships between the imported records and Elements existing users. It is strictly used to populate metadata fields like authors, contributors etc..

As an example, if we are importing publication records and we are populating the field authors, a single publication may have many authors and again there will be an entry for each author with the same publication id from the metadata.csv or a single entry with all author names combined, depending on the format we choose..

When creating the Persons file, you can choose one of the formats below:

1. A format where we have a line for each person name with additional information related to that person:
    - **first-name**: the first name of a person (e.g. *John*)
    - **surname**: the surname of a person (e.g. *Smith*)
    - **full**: the full name of a person (e.g. *Smith, J*)
    - **order-number**: the order in which this name should appear, has to be an integer (e.g. *5*)

(NOTE that if order-number is not supplied, people will be added in the same order that their rows are provided in this file).

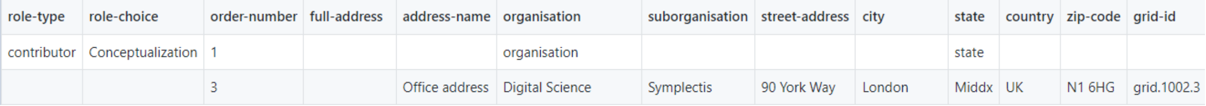
- **role-type**: the role type (e.g. contributor). See Appendix E for available values
- **role-choice**: the actual role part of the selected type above e.g Conceptualization.

- **role-type-choices**: this column has to be used when we need to provide multiple role type:choices per person. The format is:

  "creative:Illustrator|creative:Lighting Designer"

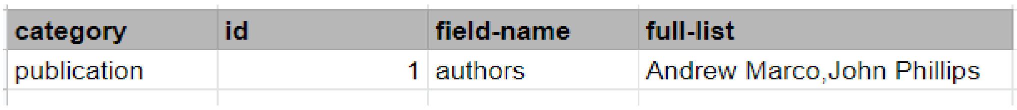
where each pair is separated by | and in each pair, the type and choice are separated by :

- **author-types:** values separated by pipe: first|last|corresponding
- **email-address:** a single valid email address
- **phone-number:** a single phone number
- **full-address**: full address of a person
- **address-name**: identifier for the address (e.g Office address)
- **organisation**: name of the organisation
- **suborganisation**: name of the suborganisation
- **street-address:** street address of the organisation
- **city:** the city where the organisation is located
- **state**: state where the organisation is located
- **country:** country where the organisation is located
- **zip-code:** zip code of the organisation
- **grid-id:** Global Research Identifier Database (e.g. grid.1002.3)
- **orcid**: ORCID researcher identifier



role-type	role-choice	order-number	full-address	address-name	organisation	suborganisation	street-address	city	state	country	zip-code	grid-id
contributor	Conceptualization	1			organisation				state			
		3		Office address	Digital Science	Symplectis	90 York Way	London	Middx	UK	N1 6HG	grid.1002.3

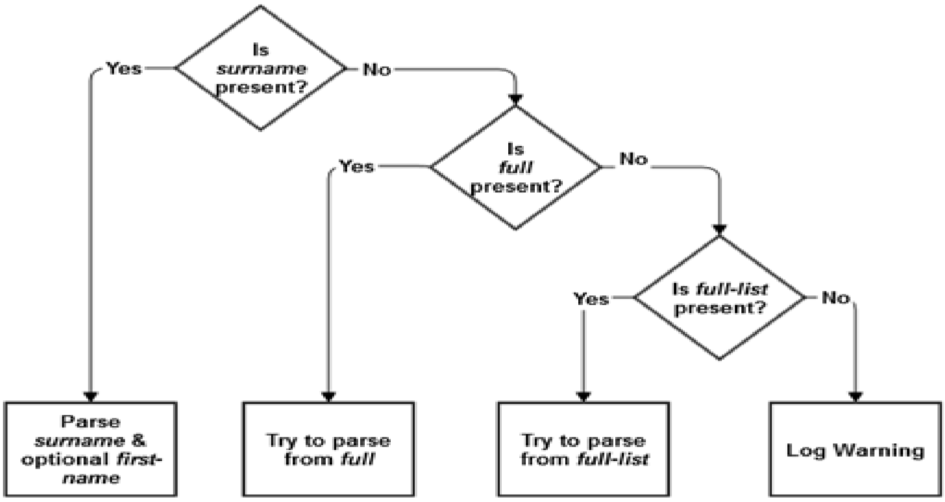
2. In the second format you can add all the person names in a single column separated by comma, with no additionalinfo per person, just the names:
   - **full-list**: a list of full names to be parsed by the importer (e.g. *Smith J, Torhvik L, and Kane C*)



category	id	field-name	full-list
publication	1	authors	Andrew Marco,John Phillips

Person Names

Ideally every person will have a first-name and a surname present in the file however, this is not always possible depending on the source of the metadata. In some cases the names may be the concatenated full name or list of names. In addition both instances can occur within the persons.csv file so the Importer attempts to retrieve something useful. The following flowchart describes the decision tree for this process:



Decision Tree for parsing User Name Fields

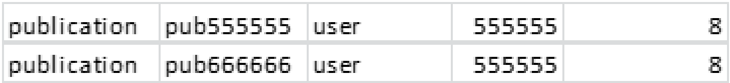
Links CSV

This file is used in the second step, after a record is created, SEI will try to create the relationships between the object that contain our imported record and other existing objects in Elements. Usually the other objects are existing Elements users, so these objects that contain the imported records would be linked to the user's profiles, but SEI can create links to any other objects, even to objects of the records imported previously withing the same batch(same metadata file).

The links.csv file has five required headers, when the other object that we want to link is an user:

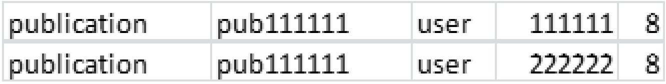
- **category-1:** An *Elements* category ie publication
- **id-1:** Unique id from the metadata.csv
- **category-2:** An *Elements* user
- **id-2:** A unique user id from hr.csv file
- **link-type-id:** Defines the type of link such as an author(See Appendix D of full list)
- **visible:** Controls the link visibility(privacy). Available values: TRUE, FALSE, YES, NO 0,1. If false, the privacy of the relationship is set to 'Private', if true, the privacy os the relationship is set to 'Public'.
- **favourite(optional):** can set if the object that contain the imported record is favorite or not for th linked user. It is relevant only for user link types. Available values: TRUE, FALSE , YES, NO 0,1.

As a category such as a publication can have more than one author(person) and an author(person) can have more than one publication a record is created in the file for each.



publication	pub555555	user	555555	8
publication	pub666666	user	555555	8

EXAMPLE: User 555555 has two publications.



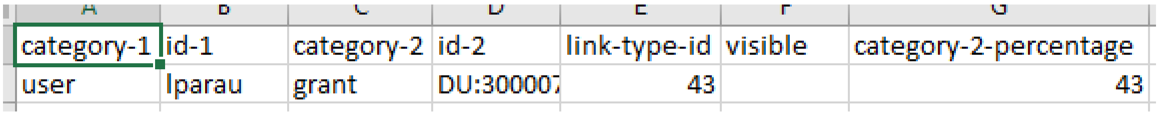
publication	pub111111	user	111111	8
publication	pub111111	user	222222	8

EXAMPLE: Publication pub111111 has two authors.

<u>Percentage can be applied only to user links.</u>

How to use the percentage columns:
Depending on which category we have the user in the links file(1 or 2) we need to specify the percentage for the other category.

- if the user is in **category-1** column, we need a new column named **category-2-percentage** ,
- if the user is in the **category-2** column, we need a new column named **category-1-percentage**.



A	B	C	D	E	F	G
category-1	id-1	category-2	id-2	link-type-id	visible	category-2-percentage
user	lparau	grant	DU:300007	43		43

Important note: The SEI does not support links synchronization. This means that older links that are no longer present in the file, or links added by users through the UI, will not be deleted in Elements. If an author is linked to an item erroneously on a first iteration, we will have to remove the link from Elements UI or use the API with a different process.

The links file can be entirely empty if we do not need to create relationships between the imported records and other objects and SEI is configured to allow orphan objects import.

2. Configure and running the import tool

IMPORTANT: The SEI will be hosted, configured, and executed on DS servers for hosted clients.

The self hosted client will use the .exe file to start the tool.

In order to perform the import, the import tool needs to have the following information:

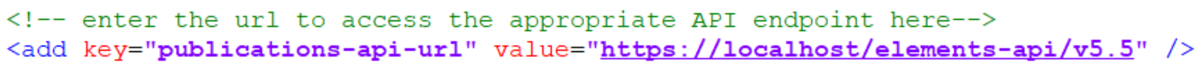
1. Path to the three source files (Person.csv, Metadata.csv, Links.csv)
2. API URL and any necessary API access credentials (for https endpoints)Source Files

The default setting for location is to expect the CSV files to be in the directory stated above and there should be no reason to change this however it can be done if required.

```
<!--The publication data to be imported-->
<add key="metadata-data-file" value=".\Data\Metadata.csv"/>
<!--The person data associated with these publications-->
 <add key="person-data-file" value=".\Data\Persons.csv"/>
<!--The links to internal users present in the HR Elements feed-->
<add key="link-data-file" value=".\Data\Links.csv"/>
<!--The source the data will be shown as in Elements-->
```

API Endpoint

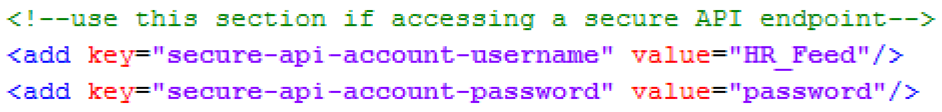
The Standard Elements Importer interacts with Elements via the API; therefore in order for it to connect it needs to know the endpoint address to access the API endpoint. Api endpoint 6.13 is supported in the latest SEI version. The API endpoint is specified in the configuration section below:



```
<!-- enter the url to access the appropriate API endpoint here-->
<add key="publications-api-url" value="https://localhost/elements-api/v5.5" />
```

The port selected will depend on the environment it is running in so this one is just for demonstration. This API must be defined in *Elements* and the API must be running.

If you are using a secure endpoint for the import, the username and password must be specified:

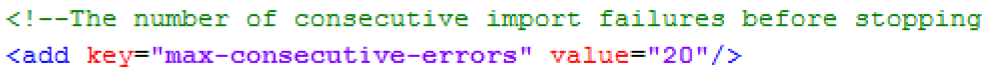


```
<!--use this section if accessing a secure API endpoint-->
<add key="secure-api-account-username" value="HR_Feed"/>
<add key="secure-api-account-password" value="password"/>
```

Additional Settings

Consecutive Errors

To allow you to effectively resolve import errors reported by the import tool, a failsafe feature has been added which will halt the import process should a specified number of consecutive errors occur. By default, the Import tool will stop after 20 consecutive errors, however, you can configure this value in the key:

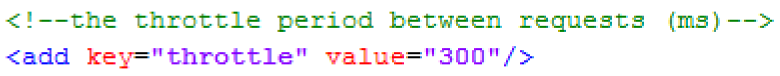


```
<!--The number of consecutive import failures before stopping
<add key="max-consecutive-errors" value="20"/>
```

If you choose to omit this key from the configuration file, the tool will still use the default value of 20.

Throttling API Requests

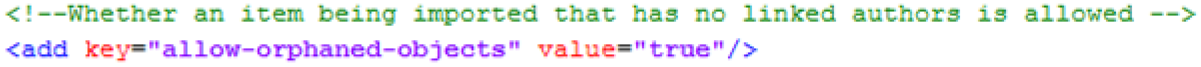
The configuration file provides the option for adjusting the throttling period between consecutive requests to the Elements API. Reducing this period will increase the upload speed, however it will place more strain on the Elements system so caution should be taken when adjusting this. The throttle period is specified in milliseconds:



```
<!--the throttle period between requests (ms)-->
<add key="throttle" value="300"/>
```

Allow Orphans

‘Orphans’ are metadata records without any associated links(relationships) specified in the Links CSV file. These can be imported as long as this is set to be allowed in the configuration:



```
<!--Whether an item being imported that has no linked authors is allowed -->
<add key="allow-orphaned-objects" value="true"/>
```

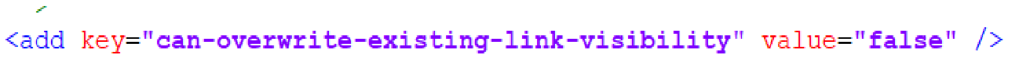
Overwrite existing links visibility

This parameter controls whether we may override the visibility of existing user links using the values provided in the (optional) column **'visible'** from links file.

If true, we set the visibility based on import file values, overriding any previous feed iterations or any manual updates.

If false, or if the key is not present, the importer will not override the existing links visibility even if values are present in the links file.

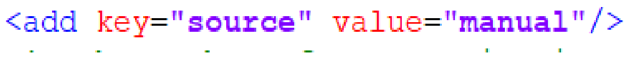
In case this key is set to True but in links import file the column named 'visible' is missing, the visibility for all existing links will be overridden to true.



```
<add key="can-overwrite-existing-link-visibility" value="false" />
```

Source

The source used to import the records in Elements. If the manual source is used, the imported records will be editable in Elements. See Appendix F for available sources for each category.

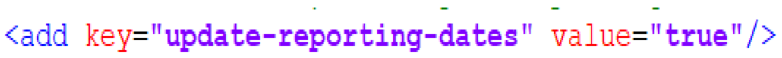


```
<add key="source" value="manual"/>
```

Available sources are in Appendix F.

Update reporting dates

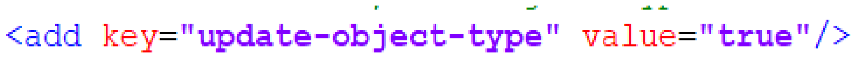
If set to true, the object reporting date will be updated based on the dates provided in the metadata file.



```
<add key="update-reporting-dates" value="true"/>
```

Update object type

If set to true, the object type will be updated based on the type column from the metadata file.



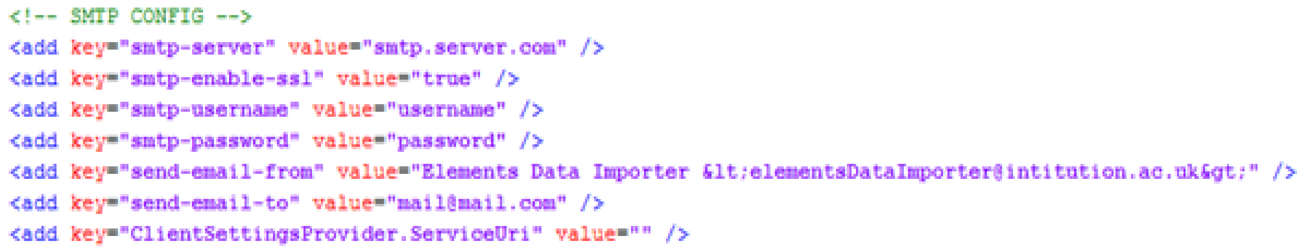
```
<add key="update-object-type" value="true"/>
```

Email configuration – SMTP

Once the Import Tool has completed, or in the event of an error, the import tool can send an email to a specified account detailed in this section. We can send email in case of failure or in case of success. In order to receive email for success or failure, you need to set the values to ‘true’ for the following lines. You can also disable these options by setting the value to ‘false’.

```
<!-- EMAIL CONFIG -->
<!--Whether to send on failure or not. -->
<add key="send-email-on-failure" value="true"/>
<!-- Whether to send an email on success or not. -->
<add key="send-email-on-success" value="true"/>
```

The SMTP configuration is set in the following keys.



```
<!-- SMTP CONFIG -->
<add key="smtp-server" value="smtp.server.com" />
<add key="smtp-enable-ssl" value="true" />
<add key="smtp-username" value="username" />
<add key="smtp-password" value="password" />
<add key="send-email-from" value="Elements Data Importer &lt;elementsDataImporter@intitution.ac.uk&gt;" />
<add key="send-email-to" value="mail@mail.com" />
<add key="ClientSettingsProvider.ServiceUri" value="" />
```

3. Running the tool

IMPORTANT: Hosted clients cannot trigger the tool but an automatic schedule will be agreed.

The clients will receive SFTP credentials where the files have to be uploaded and the tool will process the field automatically based on the agreed schedule.

Logging

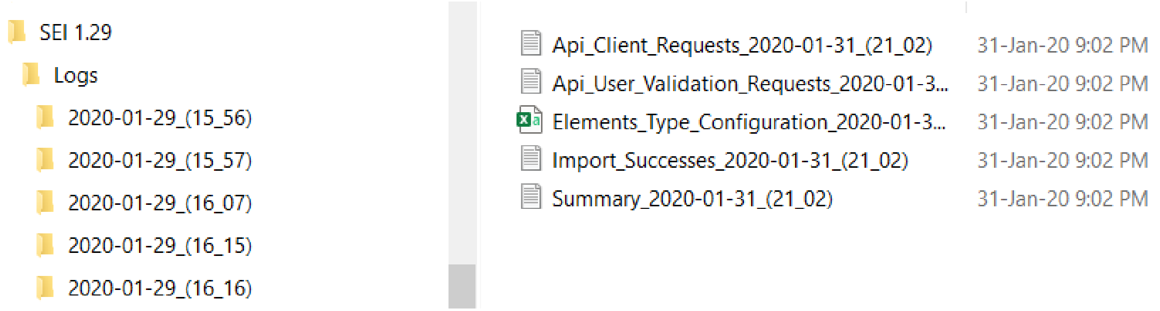
The import tool produces up to seven logs during execution; these can be used for review and troubleshooting purposes.

IMPORTANT: Hosted clients will have access to the logs.

Log name	Type	**Information**
Api_Client_Requests	Review / Troubleshooting	Logs the requests made to the API and the responses received back. Useful to review API activity and to troubleshoot API related issues.
Api_User_Validation_Requests	Review / Troubleshooting	Logs if the users from the Links file were found in Elements. This validation is performed if only the field “allow-orphaned-objects” is set to false.
Elements_Type_Configuration	Review	Provides a matrix of Elements *fields* against each *type* defined in the Elements instance begin imported into.
Import_Errors	Troubleshooting	Produced only if the import process encounters problems during execution. In order to resolve any issues with the source data, this log is a good place to start reviewing after the import process completes
Import_Successes	Review	Logs all successfully imported data.
Import_Warnings	Troubleshooting	Logs problems associated when attempting to link imports of data. Produced only if there are warnings to report

Program_Exceptions	Troubleshooting	Produced when the importer stops execution after it encounters a critical error or reaches a maximum number of consecutive errors.
Summary	Review	Provides a statistical breakdown of the items imported into Elements.

Logs are time-stamped and stored in time-stamped folders. A new folder is generated each time the import tool is run:



SEI 1.29
- Logs
  - 2020-01-29_(15_56)
  - 2020-01-29_(15_57)
  - 2020-01-29_(16_07)
  - 2020-01-29_(16_15)
  - 2020-01-29_(16_16)

Api_Client_Requests_2020-01-31_(21_02)	31-Jan-20 9:02 PM
Api_User_Validation_Requests_2020-01-3...	31-Jan-20 9:02 PM
Elements_Type_Configuration_2020-01-3...	31-Jan-20 9:02 PM
Import_Successes_2020-01-31_(21_02)	31-Jan-20 9:02 PM
Summary_2020-01-31_(21_02)	31-Jan-20 9:02 PM

Appendices

Appendix A: Categories and their default types

Category	Category Type(s)		
publication	artefact book chapter composition conference dataset	design exhibition internet-publication journal-article other patent	performance poster report scholarly-edition software thesis-dissertation
grant	grant		
activity	non-research-presentation committee-membership offices-held event-administration editorial community-service event-participation	membership employee-supervision consulting-advisory institutional-review expert-witness journal-reviewing-refereeing conference-reviewing-refereeing	grant-application-assessment promotion-tenure-assessment broadcast-interview text-interview distinction fellowship
org-structure	organisational-structure		
project	project		
equipment	instrument database service software		
teaching-activity	course-taught course-developed program-developed course-based-degree-supervision graduate-examination mentoring research-based-degree-supervision		

Appendix B: *Field* Types

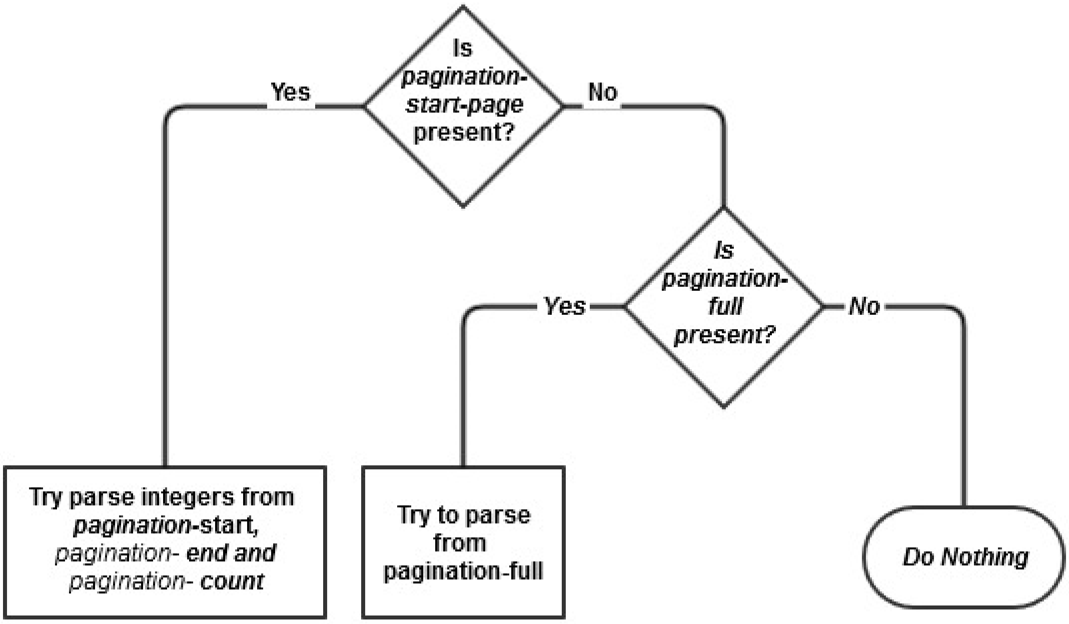
Each *field* has an underlying data type that *Elements* expects a chunk of metadata to conform to. The importer will attempt to parse each data type in a way appropriate to its type, so it will expect and respond to a text data type differently to a numerical data type. Underlying data types can take the following forms:

- Single *fields* with a single values such as a 'number'
- Lists of values concatenated together such as an 'address-list'
- Made up of multiple *fields* such as 'money'

Field Type	Description	CSV File	List Allowed	Multi Fields
text	Simple text value.	Metadata	No	No
html-text	Data can contain HTML elements that can be used to format the display of the value In Elements. i.e. "<b>This is in bold</b>, this is not."	Metadata	No	No
date	The importer will attempt to parse this data into a date type, it can do this from recognised date formats.It is indicated to use ISO date format: YYYY-MM-DD.	Metadata	No	No
choice	The data in this cell must be one of the available choices configured in Elements for this field.	Metadata	No	No
integer	The data in this cell must be a valid integer.	Metadata	No	No
boolean	The data in this cell must be one of the following to parse as true – *true*, *yes*, *y*, *1* or to parse as false – *false*, *no*, *n*, *0*.	Metadata	No	No
number	The data in this cell must be a valid decimal number.	Metadata	No	No
list	The data in this cell should be a semicolon separated list of values, each of the separated values will be parsed into a separate entry in the final list field.	Metadata	Yes	No
isbn-10	The data in this cell should be a valid isbn-10.	Metadata	No	No
isbn-13	The data in this cell should be a valid isbn-13.	Metadata	No	No
issn	The data in this cell should be a valid issn.	Metadata	No	No

person-list	A person-list field can hold multiple person names and for each person from the list, additional information can be added like address or role and this field can also be resolved to a link if the person name is recognized as an Elements user, depending of how the field is configured when added to an Elements type	Persons	Yes (If **full-list** Is used)	No
person	Again to be used in Persons.csv as opposed to a column. In this case the order number and the full-list columns will be ignored.	Persons	No	No
url	The data in this cell should be a valid URL.	Metadata	No	No
keyword-list	As for the list, the data in this cell should be a semi-colon separated list of values, each of these values will be parsed into a separate entry in the keyword list and the imported labels will be Unclassified. If the keyword list field is configured to use a specific label scheme, the importer can target it if the values are provided in this format: "label value|label-scheme-identifier|percentage" The pares can be separated by a semi-colonto import multiple labels.	Metadata	Yes	No
doi	The data in this cell should be a valid DOI.	Metadata	No	No
pagination	Pagination is multi-fielded so the Importer attempts to parse the field values based on the presence of a particular field. The diagram following this table defines the flow chart. pagination-start-page (Example 1) pagination-end-page (Example 100) pagination-count (Example 100) pagination-full (Example 1-100:100)	Metadata	No	Yes
address-list	Full addresses should be separated by a semi colon, as for lists and keyword lists, and each value will create a new address entry in the list.	Metadata	Yes	No

Identifier-list	A semi-colon separated list of identifiers. Each identifier must be enclosed in **single** quotes, any quotes within an identifier must be escaped with a **backslash**. An identifier consists of both a scheme and a value, these are separated by a colon. If a value contains colon it does not need to be escaped, since the identifier will only be split on the first identifier. An identifier scheme cannot contain a colon. An identifier scheme must be one of the following values (meaning in brackets): ● pmc (PubMed Central ID) ● arxiv (arXiv ID) ● pubmed (PubMed ID) ● nihms (NIH Manuscript Submission ID) ● isidoc (Thomson Reuters Document Solution ID) Below is an example of a possible value of an identifiers field: 'arxiv:quant-ph/0612120';'pmc:PMC3348095';'pubmed:22547652 '	Metadata	Yes	No
funding-acknowledgements	The value can contain acknowledgement-text and pairs of grant-id:organisation. The acknowledgement-text needs to be separated with a '|' from the rest of the text. The grant ID will be separated with a ':' from organisation name value. The pairs grant-id:organisation will be separated by ';'. To have a valid funding entry, both grant-id and organisation needs to be present. The value for acknowledgement-text is optional. Example: acknowledgement-text|grant-id:organisation;grant-id:organisation	Metadata	Yes	No
money	Two columns are used to specify a money field. The first is the integer amount which has the column header *–value* prefixed by the field name e.g. ***c-money-1****-value*. The second is the currency of the money and has the header *–currency-code* prefixed by the field name e.g. ***c-money-1****-currency-code*. The currency column can only contain one of the following values: AUD, BRL, CAD, CHF, CNY, EUR, GBP, HKD, INR, JPY, KRW, MXN, NOK, NZD, RUB, SEK, SGD, TRY, USD, ZAR	Metadata	No	Yes


Is pagination-start-page present?
Yes
No
Is pagination-full present?
Yes
No
Try parse integers from pagination-start, pagination- end and pagination- count
Try to parse from pagination-full
Do Nothing

Pagination Flowchart Diagram

Appendix C: Example default fields (types) by category

See also the **Elements_Type_Configuration** log

underlying field	type	publication	grant	equipment	Project
abstract	text	x	x		
additional-name	text			x	

addresses	address-list	x		x	
algorithm	text			x	
amount	number		x		
application-date	date		x		
arxiv-pdf-url	url	x			
associated-authors	person-list	x			
authors	person-list	x			
author-url	url	x			
award-date	date		x		
coded-in	text			x	
commissioning-body	text	x			
confidential	boolean	x			
contacts	person-list			x	
currency	text		x		
data-input	text			x	
data-output	text			x	
department	text (100)		x		
description	text		x	x	x
developed-by	person-list			x	
discipline	text (100)		x		

doi	doi	x			
edition	text	x			
editors	person-list	x			
eissn	issn	x			
end-date	date		x		
fee-for-service	boolean			X	
filed-date	date	x			
finish-date	date	x		X	
funder-name	text		x		
funder-reference	text		x		
funder-type	text		x		
funding-acknowledgements	funding-acknowledgements	x			
funding-type	text		x		
institution	text (200)		x		
institution-reference	text		x		
inventory-number	text			X	
is-application	boolean		x		
isbn-10	isbn-10	x			
isbn-13	isbn-13	x			
issn	issn	x			

issue	text	x			
journal	text	x			
keywords	keyword-list	x			
language	Text	x			
licence	Text			x	
location	Text	x			
manufacturers	address-list			x	
medium	Text	x			
model-number	text			x	
name	text			x	x
name-of-conference	text	x			
notes	text	x			
number	text	x			
number-of-pieces	text	x			
operating-system	text			x	
pagination	pagination	x			
parent-title	text	x			
patent-number	text	x			
patent-status	text	x			
pii	text	x			

place-of-publication	text	x			
programme	text (100)		x		
protocol	text			x	
publication-date	date	x			
publication-status	choice	x			
publisher	text	x			
publisher-url	url	x			
purpose	text			x	
references	list	x			
restrictions	list			x	
scheme	text (250)		x		
series	text	x			
service-fee-url	url			x	
start-date	date	x	x	x	
status	text		x		
sterling-value	integer		x		
sub-programme	text (100)		x		
sub-type	text			x	
thesis-type	choice	x			
title	text	x	x		

types	list	x			
url	url			x	
version	text	x		x	
volume	text	x			

Appendix D: Category:User link Types

Category	Link type ID	Link Type
publication	8	Author of
	9	Editor of
	82	Translator of
	92	Contributor to
grant	17	Funder of
	43	Primary investigator
	44	Secondary investigator
	95	Primary investigator (sub-project)
	96	Secondary investigator (sub-project)
	97	Senior/Key Personnel
	133	Has clinical evaluator

	134	Has multi-PI
	135	Has mentor
	136	Has program co-ordinator
	137	Has program director
	138	Has researcher
	139	Has statistician
	140	Is sponsored by
	141	Has other significant contributor
	116	Has co-primary investigator
	117	Has personnel member
	118	Has project lead
	119	Has co-leader
	120	Has principal investigator
	121	Has co-principal investigator
	122	Has co-investigator
	123	Has site PI
	124	Has site investigator
	125	Has consultant
	126	Has collaborator
activity	23	Associated with

org-structure	29	Director of
	30	Administrator of
	31	Member of
	45	Researcher in
	60	Head of
	61	HR manager of
	62	Safety manager of
	63	Web manager of
	64	Finance manager of
	65	Deputy head of
	66	Fellow of
	78	Board member of
	79	Committee member of
	80	Consultant for
	81	Visiting academic in
project	37	Funder of
	46	Administrator of
	47	Project manager of
	48	Championed by
	49	Lead by

	50	Team member of
	51	Researcher on
equipment	41	Used by
	52	Trained on
	67	Member of team
	68	Maintainer of
	69	Manager of
Teaching Activity	83	Associated with

Appendix E: Available person roles

Can be used in Persons file on columns role-type and role-choice starting from Elements 5.13.

role-choice	choice
contributor	Conceptualization
contributor	Data Curation
contributor	Formal Analysis
contributor	Funding Acquisition
contributor	Investigation
contributor	Methodology
contributor	Project Administration
contributor	Resources
contributor	Software

contributor	Supervision
contributor	Validation
contributor	Visualization
contributor	Writing – Original Draft
contributor	Writing – Review & Editing
creative	Actor
creative	Animator
creative	Artist
creative	Calligrapher
creative	Choreographer
creative	Cinematographer
creative	Composer
creative	Conductor
creative	Costume Designer
creative	Curator
creative	Dancer
creative	Designer
creative	Director
creative	Exhibitor
creative	Film Editor
creative	Illustrator

creative	Instrumentalist
creative	Librettist
creative	Lighting Designer
creative	Lyricist
creative	Musician
creative	Other
creative	Performer
creative	Photographer
creative	Printmaker
creative	Producer
creative	Production Personnel
creative	Programmer
creative	Recording Engineer
creative	Screenplay Author
creative	Screenwriter
creative	Set Designer
creative	Singer
creative	Sound Designer
creative	Videographer
creative	Vocalist

Appendix F: Available sources used to import data with SEI for each supported category

activity	publication	grant	project	org-structure	teaching-activity
manual	manual	manual	manual	manual	manual
c-inst-1	c-inst-1	source-3	c-inst-1	c-inst-1	c-inst-1
	c-inst-2				